ADVANCES IN INTELLIGENT SYSTEM AND COMPUTING

Volume No. 742

Issue No. 3

September - December2025

ENRICHED PUBLICATIONS PVT. LTD

S-9, IInd FLOOR, MLU POCKET,
MANISH ABHINAV PLAZA-II, ABOVE FEDERAL BANK,
PLOT NO-5, SECTOR-5, DWARKA, NEW DELHI, INDIA-110075,
PHONE: - + (91)-(11)-47026006

Advances In Intelligent System And computing

About the Journal

The Advances In Intelligent System And Computing is a peer-reviewed journal aimed at providing a platform for researchers to showcase and disseminate high-quality research in the domain of modern agriculture and plant science. With the introduction of new modern paradigms such as plant biotechnology and chemical review,IJMA promises to be a high-quality dissemination forum for new ideas, technology focus, research results and discussions on the evolution of agriculture. showcase and disseminate high-quality research in the domain of modern agriculture and plant science. With the introduction of new modern paradigms such as plant biotechnology and chemical review,IJMA promises to be a high-quality dissemination forum for new ideas, technology focus, research results and discussions on the evolution of agriculture.

ADVANCES IN INTELLIGENT SYSTEM AND COMPUTING

Editorial Team

ADVANCES IN INTELLIGENT SYSTEM AND COMPUTING

(Volume No. 742, Issue No. 3, Sep - Dec 2025)

Contents

ATwo-Stage Framework for Directed Hypergraph Link Prediction

Guanchen Xiao 1 Jinzhi Liao 1, Zhen Tan 1 Xiaonan Zhang 2 and Xiang Zhao *
Science and Technology on Information Systems Engineering Laboratory, National University of Defense Technology, Changsha 410073, ChinaHarbin Flight Academy, Harbin 150000, China
Correspondence:

ABSTRACT

Hypergraphs, as a special type of graph, can be leveraged to better model relationships among multiple entities. In this article, we focus on the task of hyperlink prediction in directed hypergraphs, which finds a wide spectrum of applications in knowledge graphs, chem-informatics, bio-informatics, etc. Existing methods handling the task overlook the order constraints of the hyperlink's direction and fail to exploit features of all entities covered by a hyperlink. To make up for the deficiency, we present a performant pipelined model, i.e., a two-stage framework for directed hyperlink prediction method (TF-DHP), which equally considers the entity's contribution to the form of hyperlinks, and emphasizes not only the fixed order between two parts but also the randomness inside each part. The TF-DHP incorporates two tailored modules: a Tucker decomposition-based module for hyperlink prediction, and a BiLSTM-based module for direction inference. Extensive experiments on benchmarks—WikiPeople, JF17K, and ReVerb15K—demonstrate the effectiveness and universality of our TF-DHP model, leading to state-of-the-art performance.

Keywords: *hyperlink prediction; hypergraph; Tucker decomposition*

Introduction

Link prediction benefits in amplifying the relations in graph-structured data [1], arousing interest from both academia and industries. Existing research mainly focuses on simple graphs where a link (also known as a relation) associates with two entities (also known as an entity), while some real-world relations consist of more than two entities, such as chemical reactions [2], co-authorship relations [3], and social networks [4], etc. As shown in Figure 1, the "Located In" relation contains NYC, New York City, The Big Apple, USA, and The United States, as follows:

$$\texttt{NYC, New York City, The Big Apple} \xrightarrow{\textit{Located In}} \texttt{USA, The United States.}$$

Thus, a hyperlink is coined to model such relations, and the graph comprised of hyperlinks is defined as a hypergraph [5]. As the relations among entities are sophisticated, the construct of a hypergraph is time-consuming and hence expensive, making its incompleteness more severe than a simple graph. To mitigate the problem, a hyperlink prediction task is introduced to facilitate the research [6]. Similar to the goal of link prediction in simple graphs, the task tries to complete the missing hyperlinks in a given hypergraph.

Example 1. Consider the bottom ellipse in green in Figure 1, given several entities, e.g., NYC, New York City, The Big Apple, USA, The United States; the target of the hyperlink prediction is to determine whether there is a hyperlink and what it is (i.e., "Located In") once existing.

Thus, machine should also acquire the ability to predict the direction of the hyperlink to form the final

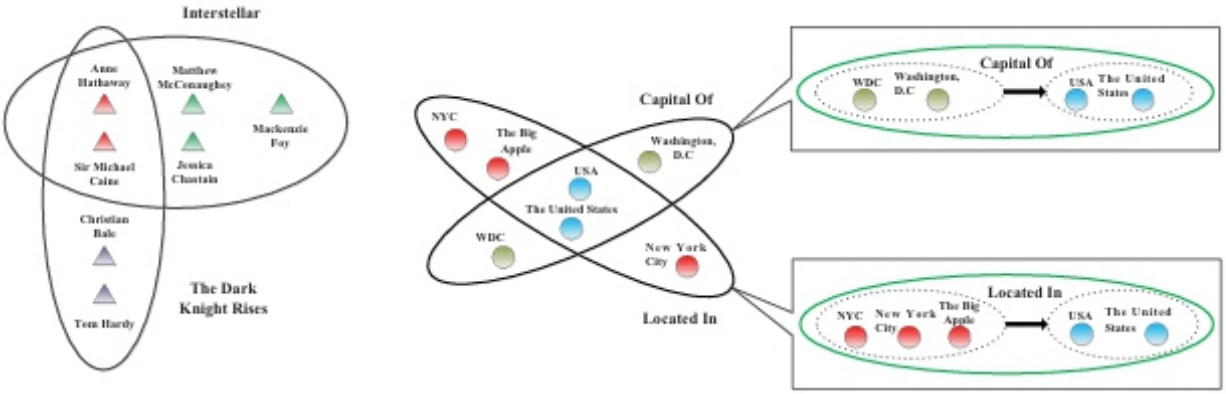


Figure 1. Sketch of two types of hypergraphs. The diagram on the left represents an undirected hypergraph while the diagram on the right stands for a directed hypergraph. One ellipse denotes a hyperlink. entities in the same ellipse share the same hyperlink. Arrow denotes the direction of the hyperlink.

To approach this task, current studies mainly fall into two categories: (1) Translation-based models try to generalize the translation constraint in simple graphs to hypergraphs, e.g., m-TransH [7], RAE [8], and NHP [9]. m-TransH directly extends TransH [10] for binary relations to the n-ary case, and RAE further integrates m-TransH with multi-layer perceptron (MLP) by considering the relatedness of entities. Since they use the sum after the projection as the scoring function, when some entities in a hyperlink change, it may not be obvious in the scoring function. (2) Neural-network-based models exploit structural information of hypergraphs, e.g., NaLP [11], HGNN [12], and HyperGCN [13]. These methods design some graph neural networks (GNNs) to absorb neighbouring features to improve entities' representations. As GNNs usually incorporate a large number of parameters, the sufficient learning process relies on the amount of training samples.

Albeit attracting attention, hyperlink prediction is still notoriously challenging, since existing studies neglect the cores of the task. First, sometimes, the accurate record of facts in a hypergraph necessitates the direction of hyperlinks. For a directed hyperlink, the entities can be divided into two parts—head and tail—based on the hyperlink's direction. This mandates that the *order* of the two parts matters; in contrast, the specific order in each part is insignificant. As shown in Figure 1, without the arrow pointing, we cannot figure out how these entities construct the relation "Located In". In addition, `NYC`, `New York City`, and `The Big Apple` (also known as the head) should be in front of `USA` and `The United States` (also known as tail), but the order inside the head or tail does not affect the determination. Nevertheless, existing methods mainly focus on undirected hyperlinks. The only method, namely, NHP, tries to average the entity embeddings generated by GCN [14] to calculate a score for inferring the hyperlink direction, which is too rudimentary to embody the direction's features. Second, as a hyperlink contains more than two entities, each entity contributes to the final existence prediction. In this light, a good representation model needs to consider the representation of all the individual entities involved in a hyperlink when making a determination. However, the current treatment of embedding tends to apply a simple sum or average strategy. This might be insensitive to the number of entities in a hyperlink since an entity with effusive containment could overwhelm other entities' expressions. Last but not least, as it is sometimes complicated for even a human being to annotate hyperlinks, there is a lack of training data, which can be currently insufficient to train a large number of learnable parameters well.

In order to address these challenges, we propose a simple yet effective model, which is a Two-stage Framework for Directed Hyperlink Prediction, namely, TF-DHP. The model is expected to equally consider the entity's contribution to the form of hyperlinks and emphasize not only the fixed order between two parts but also the randomness inside each part. It conceives a pipeline of two tailored modules: a Tucker decomposition-based module for hyperlink prediction and a BiLSTM-based module for direction inference.

For predicting the existence of hyperlinks, we exploit *Tucker decomposition* to model hyperlinks, which, to the best of our knowledge, has not been applied to hypergraphs except simple graphs [15]. In particular, instead of applying three-order Tucker decomposition over simple graphs, we employ high-order Tucker decomposition for hypergraphs. It produces a core tensor, which represents the degree of interaction between entities. Then, we devise a scoring function by the mode product of the tensor with each entity representation, which evaluates the existence of hyperlinks. We theoretically show that the score is invariant to the order of mode product with entities, though there is a direction of each hyperlink. In addition, it is noted that the tensors from Tucker decomposition are usually of very high order, which can bring about high computational complexity. To mitigate the issue, we further introduce Tensor Ring (TR) [16] decomposition to decompose higher-order tensors into mode products of several third-order tensors, which effectively reduces the computational cost.

For inferring directions, we first recall that example in Figure 1. Once `USA` and `The United States` are determined as the tail entities, the substances in the head entities are implied, and if there is a change in one of the tail entities, the head entities are going to be different. Thus, it is of importance for the model to pass the information between the two parts both forward and backward. This motivates us to design a model that works bidirectionally. In this connection, BiLSTM [17] is utilized to serve as the base model. In addition, the position of entities within the head (or tail) part is insignificant, and hence, it is necessary to train the model to attend only to the order of the two parts. For this characteristic, we keep the order of two parts but randomly shuffle entities within each part to enforce the model to be ignorant of entity positions within head (or tail) part, while being attentive to the order between the two parts. In this way, the data scale is increased as a by-product, alleviating the lack of data.

Contribution. In summary, we make the following contributions:

- For existence prediction, we propose, among the first, to generalize Tucker decomposition to a high dimension and introduce a tensor ring algorithm to reduce the model complexity. We theoretically prove that the mode product for scoring a hyperlink is invariant of the order of participating entities.
- For direction inference, we conceive a BiLSTM-based model that can take information into consideration both forward and backward with respect to a hyperlink. A data shuffling strategy is further incorporated to enforce the model to be ignorant of entity positions within the head (or tail) part while being attentive to the order between the two parts.
- The modules constitute a new model, namely, TF-DHP for predicting directed hyperlinks. Through the experiments on several real-world datasets, we confirm the superiority of TF-DHP over state-of-the-art models.

Organization. The rest of the article is structured as follows. Section 2 introduces related work and Section 3 provides a detailed account of TF-DHP. Section 4 reports the experimental setup and analyses the experimental results. Section 5 concludes the paper.

2. Related Work

In this section, we are going to review related work in link prediction on simple graphs, undirected hypergraphs and directed hypergraphs.

2.1. Link Prediction on Simple Graph

Most of the link prediction methods on simple graphs can be divided into three categories—linear mathematics models, non-linear convolutional models, and random walk models.

There were many linear mathematics ways of link prediction created in recent years such as RESCAL [18], DistMult [19], ComplEx [20], and SimplE [21]. RESCAL, which is based on tensor factorization, performs collective learning via the latent components of the model and provides an efficient algorithm to compute the factorization. DisMult is a special case of RESCAL with a diagonal matrix per relation which reduces overfitting while ComplEx extends DisMult to the complex domain. SimplE is based on Canonical Polyadic (CP) decomposition, in which subject and object entity embeddings for the same entity are independent. TuckER [15] is a straightforward but powerful model based on the Tucker decomposition; it considers the core tensor as the parameter tensor, and the scoring function is defined by taking the modular product between the entities embedding vectors, the relation embedding vector, and the core tensor. Because the information loss in the calculation process is greatly reduced by using high-order tensors to define parameters, TuckER is proved to be the best-performing linear mathematics model to handle the link prediction task on simple graph.

Typical works of non-linear convolutional models are ConvE [22] and HypER [23]. ConvE is a simple multi-layer convolutional architecture for link prediction and is defined by a single convolution layer, a projection layer to the embedding dimension, and an inner product layer. HypER's hypernetwork generates relation-specific filters, and thus extracts relation-specific features from the subject entity embedding. It necessitates no 2D reshaping and allows entity and relation to interact more completely, rather than only around the concatenation boundary.

LRW [24], MIRW [25], and MLRW [26] are random walk-based models for link prediction on complex networks, LRW is conducted using pure random walking and selects the destination entities based on a random manner. To help to improve the LRW, the concept of asymmetric mutual influence of entities is presented, and using this concept, the walker selects the next entity using its effect on the current entity and selects more efficient paths for the next step. Therefore, entities with a more significant structural similarity will obtain a higher score in the proposed algorithm MIRW. MLRW provides a framework to extend the local random walk method to multiplex networks so that we can take advantage of intra-layer and interlayer information presented in the network and increase the accuracy of link prediction properly.

2.2. Link Prediction on Undirected Hypergraph

The general work on the undirected hyperlink prediction can be divided into two species, i.e., translation-based models and neural network-based models.

The representative model of the translation-based approaches are m-TrnasH [7] and RAE [8]. m-TransH generalizes TransH [10] to the case of n-order relations, and it projects entities onto the relation-specific hyperplane and defines the scoring function as the weighted sum of projection results. RAE considers the possibility of common occurrence between entities in n-order relations, establishes the correlation model through MLP, and reflects it in scoring function. Since these models are extended from binary models, restrictions on the representation of relations are also carried to the representation of n-order relations.

NaLP [11], HyperGCN [13], and Hyper-SAGNN [27] are three neural network-based approaches. HGNN is a general hypergraph neural network framework based on hypergraph convolution operation, which can incorporate multi-modal data and complicated data correlations. HyperGCN proposes a new method of training a GCN on hypergraph using tools from the spectral theory of hypergraphs and applying the method to the problems of SSL(hypergraph-based semi-supervised learning) and combinatorial optimization on real-world hypergraphs. Hyper-SAGNN develops a new self-attention based graph neural network applicable to homogeneous and heterogeneous hypergraphs with variable hyperlink sizes.

2.3. Link Prediction on Directed Hypergraphs

The research of link prediction on directed hypergraphs is not very mature, and most methods prefer predicting the direction of the hyperlink after finishing predicting the entities contained in the hyperlink. The NHP [9] model sets up two scoring functions to predict hyperlinks and their directions based on the GCN template, and they divide a hyperlink into two sub-hyperlinks and use their embedding vectors to compute the scoring function for direction. However, as the embedding vectors of hyperlinks are from the average value of entity embedding vectors, information about entities and their positions is lost, which makes the performance of the model barely satisfactory.

3. Method

This section formalizes the task of the directed hypergraph link prediction and presents the proposed method, including the framework and module details. Definitions of notations used in the text are shown in the Table 1.

Table 1. Descriptions of notations used in the following parts.

Symbol	Definition
χ	a kth-order tensor $\in R^{I_1 \times I_2 \times \ldots \times I_{k-1}}$
ω	a kth-order core tensor $\in R^{J_1 \times J_2 \times \ldots \times J_{k-1}}$
$\omega_{j_1 j_2 \cdots j_{k-1}}$	$(j_1, j_2, \cdots, j_{k-1})$-th element of ω
$U^{(n)}$	n-mode factor matrix $\in R^{I_n \times J_n}$
$u^n_{j_n}$	j_n-th column vector of $U^{(n)}$
$\phi(\cdot)$	scoring function of the existence of hyperlinks
r	relation embedding of hyperlink
v_m	embedding of entities
$\times_n$	tensor n-mode product
$Z_k(i_k)$	i_k-th lateral slice matrix of TR origin tensor
$Trace(\cdot)$	matrix trace operator
$\oplus$	concatenating operation for hidden layers
$\circ$	vector outer product

3.1. Task Description

A directed hypergraph is an ordered pair $H = (V, E)$, where $V = \{v_1, \ldots, v_l\}$ denotes a set of entities and l is the number of entities. E comprises a set of directed hyperlinks, formally:

$$E = \{(h_1, t_1), (h_2, t_2), \ldots, (h_m, t_m)\} \tag{1}$$

Each element in E can be divided into two components, where h (resp. t) serves as the *head* (resp. *tail*), with the direction being from the head to the tail.

The directed hyperlink prediction aims to predict the missing hyperlinks, including the existence and associated direction, based on the relevance of the given entities. Take relation knowledge in Figure 1 as an instance. Entities in each relation build the V, and their corresponding relation forms the directed hyperlinks E. Every sample in the dataset will contain an uncertain number of substances. We have to determine whether they can support a relation knowledge and which component each entity belongs to.

3.2. Framework

TF-DHP consists of a Tucker decomposition-based hypergraph link prediction model and a BiLSTM-based direction prediction model to predict directed hyperlinks among entities sets in a directed hypergraph. It is then optimized by a ranking objective in which scores of existing hyperlinks are ranked higher than those of non-existing entity subsets and scores of positive directions are higher than those of negative directions. The framework is shown in Figure 2.

We generalize TuckER [15] to the high dimension and regard it as a scoring function. We use the scoring function after obtaining the embedding vectors of every entity in an entity set to evaluate whether the hyperlink exists or not. If the hyperlink does exist, we divide the entities set into two groups based on the direction label of each entity and then use the BiLSTM model [17] to evaluate the direction between the groups which can be defined as the direction of the hyperlink. Meanwhile, we also randomly sort the entities in each group to increase training data according to the characteristic that the order of entities in each group does not influence the direction.

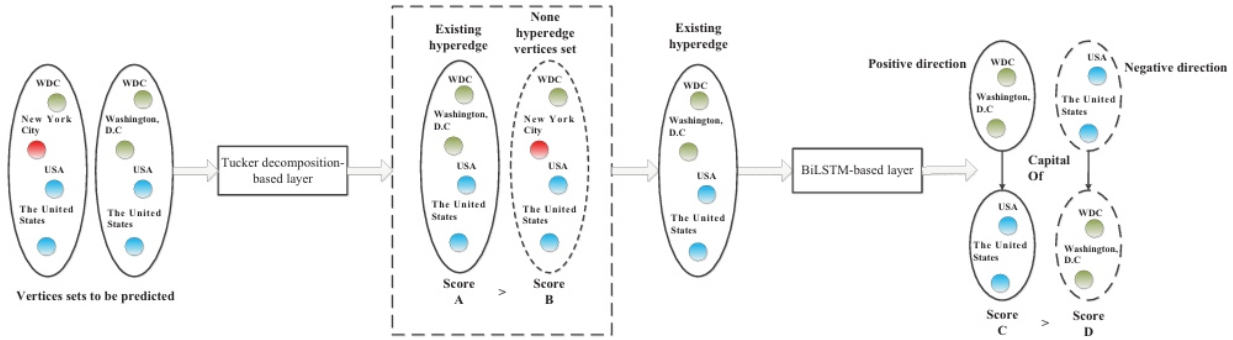


Figure 2. A sketch of TF-DHP directed hypergraph prediction model. The embedding of entity sets to be predicted are fed into the Tucker-decomposition-based layer to calculate the score. The target of model training is to make the score of existing hyperlinks larger than the score of entities set without hyperlinks. Then, the embeddings of entities in the existing hyperlink are sent to the BiLSTM layer to calculate the direction score. The target of model training is to make the score in the positive direction larger than the score in the negative direction.

3.3. Tucker Decomposition-Based Hyperlink Prediction Module

To predict hyperlinks of the entity set, we propose a Tucker decomposition-based scoring function and provide mathematical proof of its irrelevance with the order of inputs.

3.3.1. Tucker Decomposition-Based Scoring Function

Tucker decomposition is a tensor decomposition algorithm that decomposes higher-order tensors into a core tensor and several factor matrices. The core tensor reflects the degree of interaction between different factor matrices. The formal expression is as follows:

$$\chi = \omega \times_1 U^{(1)} \times_2 U^{(2)} \cdots \times_{k-1} U^{(k-1)} = \sum_{j_1=1}^{J_1} \sum_{j_2=1}^{J_2} \cdots \sum_{j_{k-1}=1}^{J_n} \omega_{j_1 j_2 \cdots j_{k-1}} u_{j_1}^{(1)} u_{j_2}^{(2)} \cdots u_{j_{k-1}}^{(k-1)} \quad (2)$$

where $\mathcal{X} \in R^{I_1 \times I_2 \times \ldots \times I_{k-1}}$ denotes the original tensor, $\omega \in R^{J_1 \times J_2 \times \ldots \times J_{k-1}}$ denotes the core tensor and $J_1 J_2 \cdots J_{k-1}$ are much smaller than $I_1 I_2 \cdots I_{k-1}$, k denotes the order of $\mathcal{X}$, $U^{(1)}, \ldots, U^{(k-1)}$ denotes the set of factor matrices, and the mathematical symbol $\times_k$ denotes the tensor product along with the kth mode. The dimensions of the core tensor are smaller than those of the original tensor in each order, so the core tensor can be regarded as the dimensionality reduction in the original tensor.

Based on the Tucker decomposition of the representation tensor, we design the scoring function to score each hyperlink. Specifically, if a hyperlink contains m entities, we first select the corresponding entity and relation embeddings. Then, a parameter tensor is designed as the core tensor containing learnable parameters shared by entities and relations [15]. Our goal is to optimize these parameters to fully exploit the relevance among entities and the associated relations based on their embeddings. The scoring function can be expressed as below:

$$\phi(r, v_1, v_2, \ldots, v_m) = \omega \times_1 r \times_2 v_1 \times_3 \ldots \times_{m+1} v_m, \quad (3)$$

where m changes with the number of entities contained in the hyperlink, and the order of the tensor $\mathcal{Z}$ is equal to one plus the number of entities. r denotes the relation embedding of the hyperlink to be predicted, and $v_1, v_2, \ldots, v_m$ are the embeddings of entities contained by the hyperlink. Since the tensor product of a tensor with a vector will change the dimension of its corresponding order to 1, we can repeat the process $m + 1$ times to acquire a real number. This real number is further regarded as the score of this hyperlink.

As every entity in the hyperlink and the relation embedding are computed simultaneously, Equation (3) reduces information loss. Nevertheless, the computational complexity becomes enormous with the increase in the number of entities because of the inner computation of the high-order tensor product. To address the issue, we use the TR [16] decomposition algorithm. It represents a high-order tensor by a sequence of third-order tensors multiplied circularly, mathematically:

$$T(i_1, i_2, \ldots, i_n) = Trace\{Z_1(i_1) Z_2(i_2) \cdots Z_n(i_n)\} = Trace\{\prod_{k=1}^{d} Z_k(i_k)\} \quad (4)$$

where T denotes the original tensor of size $n_1 \times n_2 \times \cdots \times n_d$, Z_k denotes a set of third-order tensors whose dimensions are $r_k \times n_k \times r_{k+1}$, i_k denotes i_k-th layer matrix in the second-order of the tensor, and Tr denotes the trace of the product of matrices. The tensor ring decomposition makes the third dimension of the last decomposed tensor the same as the first dimension of the first decomposed tensor. The advantage is that when we make a circular shifting of the decomposed tensor, the results will not be changed because of the matrix trace operation. Tensor ring decomposition dramatically reduces the computational load of the model when the tensor order is large by decomposing higher-order tensors into products of third-order tensors.

The computational complexity grows sharply when the order of the core tensor grows, so we use the TR decomposition on the core tensor to decompose the high-order tensor into several three-order tensors multiplied circularly. Based on the definition of TR decomposition, every single parameter in the core tensor can be computed by the trace of the matrices product. It can be expressed in the tensor form [16], given by:

$$Z = \sum_{\alpha_1,\ldots,\alpha_n=1}^{r_1,\ldots,r_n} Z_1(\alpha_1,\alpha_2) \circ Z_2(\alpha_2,\alpha_3) \circ \cdots \circ Z_d(\alpha_d,\alpha_1) \tag{5}$$

where $Z_i(\alpha_k,\alpha_{k+1})$ denotes the vector corresponding to the index in the tensor and the symbol $\circ$ denotes the outer product of vectors, $r_1,\ldots,r_n$ correspond to the dimension of the first and 3rd order of the tensor. We use the simplified form $Z = Tr(Z_1,Z_2,\ldots,Z_n)$ to represent the decomposition of the core tensor. Combining with Equation (3), we can rewrite the scoring function as:

$$\phi(r,v_1,v_2,\ldots,v_n) = Trace(Z_1,Z_2,\ldots,Z_n) \times_1 r \times_2 v_1 \times_3 \cdots \times_{n+1} v_n \tag{6}$$

This scoring function not only considers all the entities and relation information contained in a hyperlink but also controls the model complexity within an acceptable range. As shown in Table 2, the scoring function above has fewer parameters than NaLP and is not easy to overfit in the datasets which are not large enough, concretely shown in Figure 3.

Table 2. Scoring functions of several models for undirected hypergraph link prediction tasks, with the significant terms of their model complexity. n_e and n_r are the number of entities and relations, while d_e and d_r are the dimensionalities of entity and relation embeddings respectively. n is the number of entities in a hyperlink and d_{max} is the maximum size of TR latent tensors. *maxmin* is the element-wise difference of maximum and the minimum values of the vectors.

Model	Scoring Function	Model Complexity
RAE	$\|\Sigma_{j=1}^{n} a_j(e_{i_j} - w_{i_r}^T e_{i_j} w_{i_r}) + r_{i_r}\|_p$	$O(n_e d_e + n_r d_r)$
NaLP	$FCN_2(min(FCN_1(Conv([W_r,[e_{i_1};e_{i_2};...;e_{i_n}]]))))$	$O(n_e d_e + n n_r d_r)$
NHP-U-mean	$\sigma(\frac{1}{\|e\|} W \cdot \sum_{v\in e} h_v^{(e)} + b)$	$O(n_e d_e)$
NHP-U-maxmin	$\sigma(W \cdot maxmin\{h_v^{(e)}\}_{v\in e} + b)$	$O(\sum_{e\in E} \frac{1}{2} \cdot \|n_e\| \cdot (\|n_e\| - 1))$
TF-DHP	$Trace(Z_1,Z_2,\ldots,Z_n) \times_1 r \times_2 v_1 \times_3 \cdots \times_{n+1} v_n$	$O(n_e d_e + n_r d_r + n d_{max}^3)$

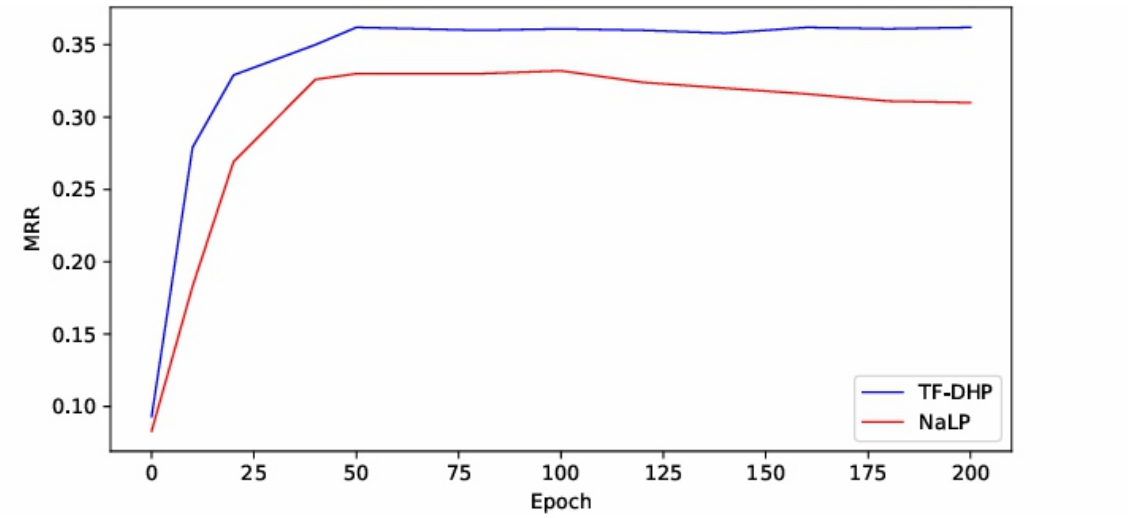


Figure 3. MRR results on NaLP and TF-DHP with training epoch growing, evaluated on WikiPeople.

This model is based on the scoring function of Tucker decomposition, and because the model needs to determine the order of the core tensor, the model cannot process the hyperlinks with different number of nodes in one time. For datasets with such hyperlinks, we need to classify them before predicting, which increases the workload to a certain extent.

As the order of the core tensors increases, the number of third-order tensors required by TR decomposition increases accordingly, which will increase the amount of computation to a certain extent. The machine used in this paper can deal with the prediction task of hyperlinks with up to six nodes.

3.3.2. Proof of Sequence Independence

As illustrated above, the Tucker decomposition processes the inputs sequentially, while the order of entities contained in one hyperlink does not influence the determination,which requires the invariance property of our scoring function. We prove that the order of entities' and relations' embeddings in the tensor product makes no difference to the result. We first rewrite the scoring function in the tensor-wise form:

$$\phi(r, v_1, v_2, \ldots, v_n) = \sum_{\alpha_1,\ldots,\alpha_n=1}^{r_1,\ldots,r_n} Z_1(\alpha_1, \alpha_2) \circ \cdots \circ Z_d(\alpha_d, \alpha_1) \times_1 r \times_2 v_1 \times_3 \cdots \times_{n+1} v_n \quad (7)$$

In the mentioned TR decomposition, the matrix trace operation and the same dimensions of the input and output ensure the invariance of circular shifting. When it comes to the hypergraph, the dimensions of entities and relations are set to a fixed value, which makes the invariance not only in circular shifting but also in order changing between every single entity. It means the change in the order of the product does not change the result. So, we just need to prove that the order of the tensor product in the Tucker decomposition has no effect on the result. The element-wise form of the tensor product is as follows:

$$\begin{aligned} \chi_{i_1 i_2 \cdots i_n} &= (\omega \times_1 U^{(1)} \times_2 U^{(2)} \cdots \times_n U^{(n)})_{i_1 i_2 \cdots i_n} \\ &= \sum_{j_1=1}^{J_1} \sum_{j_2=1}^{J_2} \cdots \sum_{j_n=1}^{J_n} \omega_{j_1 j_2 \cdots j_n} u^{(1)}_{i_1 j_1} u^{(2)}_{i_2 j_2} \cdots u^{(n)}_{i_n j_n} \end{aligned} \quad (8)$$

On the right-hand side of the equation, if we regard the indices $j_1, \ldots, j_n$ as a set of integer-independent variables and their variation range is from 1 to $J_1, \ldots, J_n$, $(u^{(1)}_{i_1 j_1}, u^{(2)}_{i_2 j_2}, \ldots, u^{(n)}_{i_n j_n})$ can be regarded as the functions of these independent variables, the meaning of the function value is the value of the element at the corresponding position in the entity embedding vector indexed by the independent variable. We use $f_1(j_1), f_2(j_2), \ldots, f_n(j_n)$ (in Equation (9)) to represent the functions. The expression $\omega_{j_1 j_2 \cdots j_n}$ can be regarded as a multivariate function whose form is $g(j_1, j_2, \ldots, j_n)$, and the value of the function means the parameter on the corresponding position of the core tensor.

Then, we find that if we make the independent variables take the value of all real numbers from 1 to J_n instead of being integers, we can transform Equation (8) into a multiple definite integral:

$$\iiint \cdots \int_D g(j_1, j_2, \ldots, j_n) f_1(j_1) f_2(j_2) \cdots f_n(j_n) dj_1 dj_2 \cdots dj_n \tag{9}$$

The integral domain D of this multiple integrals is an n-order tensor that has the same size as the core tensor. Changing the order of independent variables in $g(j_1, j_2, \ldots, j_n)$ does not change the corresponding parameter; thus, the order of $j_1, \ldots, j_n$ has no influence of the function $g(j_1, j_2, \ldots, j_n) f_1(j_1) f_2(j_2) \cdots f_n(j_n)$.

Since the functions $f_1(j_1), \ldots, f_n(j_n)$ are all unary function, the integral can be rewritten as:

$$\iiint \cdots \int_D g(j_1, j_2, \ldots, j_n) dj_1 dj_2 \cdots dj_n \int_1^{J_1} f_1(j_1) dj_1 \int_1^{J_2} f_2(j_2) dj_2 \cdots \int_1^{J_n} f_n(j_n) dj_n \tag{10}$$

For the multiple definite integrals $\iiint \cdots \int_D g(j_1, j_2, \ldots, j_n) dj_1 dj_2 \cdots dj_n$, the limit of integration for each order are finite constants, and the order of $j_1, \ldots, j_n$ makes no difference to the function, so changing the order of integration does not change the value of the definite integral. Therefore, the whole integral has the invariance property. Because Equation (8) is a special case of Equation (9), the scoring function is proven to have the invariance property.

3.4. BiLSTM-Based Direction Prediction Module

In the directed hyperlink prediction problem, the embedding of each entity further determines the existence of a hyperlink and its direction. However, different from the existence prediction, the direction of a hyperlink emphasizes the order of entities. For example, in the related knowledge "`WDC, Washington D.C` $\xrightarrow{Capital\ Of}$ `USA, The United States`", the direction comes from `WDC` and `Washington D.C` (also known as head entities) to `USA` and `The United States` (also known as tail entities). Once a substance is placed in the wrong component, the reaction might not even exist. In addition, the interaction between two components, e.g., conservation of materials, indicates that the model cannot individually determine the components. Therefore, we apply BiLSTM in our module to encode all entities sequentially to achieve the information passing both forward and backward.

As shown in the Figure 4 The BiLSTM consists of several LSTM hidden layers. These hidden layers are divided into two groups that meet end-to-end in opposite directions. The entities' embeddings in the hyperlink are calculated in the hidden layer of the corresponding position one by one. Meanwhile, the state of the previous hidden layer is calculated in the next hidden layer together with the embedding of the entities fed into the corresponding layer. After all hidden layers have been calculated, embedding containing all sequential

information is generated.The same process occurs in the backward hidden layer group, which means we can obtain two embeddings of the hyperlink. We concatenate them into one vector and then send it to a Softmax layer to obtain the direction score. The specific expression of the process is as follows:

$$\overleftarrow{h_t} = \overleftarrow{LSTM}(\overleftarrow{h_{t+1}}, w_t) \tag{11}$$

$$\overrightarrow{h_t} = \overrightarrow{LSTM}(\overrightarrow{h_{t-1}}, w_t) \tag{12}$$

$$h_t = \overrightarrow{h_t} \oplus \overleftarrow{h_t} \tag{13}$$

$$p = Softmax(h_t) \tag{14}$$

where h_t denotes the concatenated embedding of the sequential representation, $\overrightarrow{h_t}$ and $\overleftarrow{h_t}$ are calculated by two hidden layers in opposite directions, w_t denotes the embedding for the tth entity, and the symbol $\oplus$ means the concatenating operation.

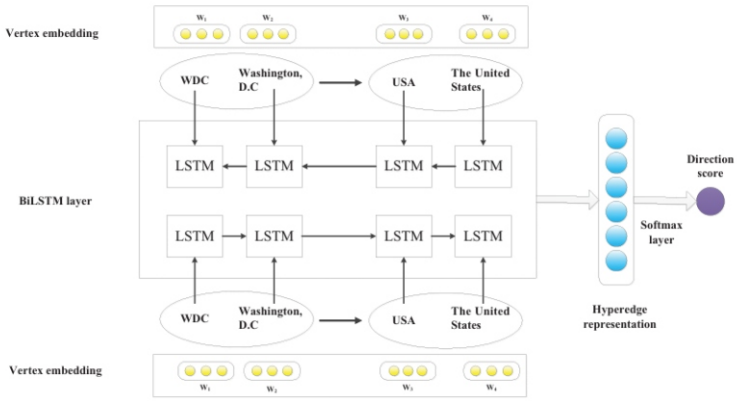


Figure 4. A sketch of the BiLSTM-based hyperlink direction prediction model, the entities in the directed hyperlink are divided into *head* and *tail* parts according to the label and are input into the BiLSTM layer in a specific order. The hyperlink representation is obtained by splicing the representation vectors obtained from each direction of the BiLSTM layer. Finally, the direction score is obtained through a Softmax layer.

As the inner order of entities in one component does not change the elements, it also has no effect on the direction, e.g., "`WDC , Washington D.C` $\overset{Capital\ Of}{\longrightarrow}$ `USA, The United States`" and "`Washington D.C , WDC` $\overset{Capital\ Of}{\longrightarrow}$ `The United States , USA`" are the same relation knowledge. However, they might be regarded as two different instances when fed into BiLSTM concentrating only on the specific sequence. In other words, if "`WDC , Washington D.C` $\overset{Capital\ Of}{\longrightarrow}$ `USA , The United States`" is annotated as the positive instance, BiLSTM cannot naturally and directly determine the correctness of "`Washington D.C , WDC` $\overset{Capital\ Of}{\longrightarrow}$ `The United States , USA`" without other guidance. Therefore, we enlighten BiLSTM to focus on the order of two components and ignore the order of entities in the same component through a data shuffling strategy. Specifically, we maintain the order of two components and randomly shuffle the entities in the same component. The number of generated instances relies on how many entities every component owns. For "`WDC , Washington D.C` $\overset{Capital\ Of}{\longrightarrow}$ `USA , The United States`", there will be $2 \times 2 = 4$ different sequences. We then

give all generated instances a correct label to enforce BiLSTM to exploit features of the direction. The strategy can enlarge the data scale without introducing external manual efforts, which also contributes to tackling the low-data regime problem.

3.5. Training

TF-DHP is a pipeline model, which means that we predict the hyperlink's existence in the first stage and judge the direction of the hyperlink in the second stage. If we use the data of undirected hypergraphs to train the first stage of the model separately, we can obtain a model that can perform link prediction of undirected hypergraphs. If the whole model is trained on the data of the directed hypergraph, the trained model can have the ability to predict directed hyperlinks.

The TF-DHP is trained in two stages, which keeps the same pace with the framework. The training goal of the first stage is to provide the existing hyperlink with a higher score while decreasing the score of entities that cannot comprise a hyperlink.With the initial embeddings of entities and their labels as input, we use the Tucker decomposition-based scoring function to obtain two kinds of the score, and a binary cross-entropy loss function is designed to maximize their gap.

After the first stage of the model is trained, we acquire the updated core tensor and embeddings and use these embeddings to initialize the second stage of the model. Two kinds of scores are calculated in the BiLSTM. One is the score of the correct direction, and the other is the score of the wrong direction. The specific expression of the loss function is as follows:

$$L = f_{mean}(\log(1 + e^{f_{mean}(\sigma(\phi_{d_n}))-\sigma(\phi_{d_p})})) \tag{15}$$

where f_{mean} denotes an average function, σ denotes the sigmoid function, ϕ_{d_n} denotes the score of each negative hyperlink, and ϕ_{d_p} denotes the score of each positive hyperlink. Finally, the BiLSTM-based model updates the model parameters and embeddings of entities and relations based on the loss gradients.

4. Experiment

This section reports the experiments.

4.1. Experimental Setup

We detail the adopted datasets, evaluation metrics, parameters, and baselines.

4.1.1. Datasets

We use two public relational datasets in our experiment for undirected hypergraph link prediction and one open KB canonicalized dataset for directed hypergraph link prediction. We brief these datasets below.

- **WikiPeople** [11]: WikiPeople is a public n-ary relational dataset concerning entities of type *human* extracted from Wikidata. WikiPeople is an incomplete hypergraph with many hyperlinks missing [11]. In WikiPeople, each set of entities has one kind of relationship. We use this dataset to train the undirected hyperlink prediction model.
- **JF17K** [8]: JF17K is a public n-ary relational dataset that has high-quality facts. It is filtered from Freebase while having multi-fold relational structures preserved. The same as WikiPeople, each set of entities has one kind of relationship, and we use this dataset to train the undirected hyperlink prediction model.

- **ReVerb15K** [9,28]: ReVerb45K is an open KB canonicalization dataset [28], and it is constructed by intersecting information from ReVerb Open KB [29], Freebase entity linking information from [30], and Clueweb09 corpus [31]. In triples of the original dataset, there may be different subjects or objects having the same meaning. Based on the Freebase entity linking information, we cluster the synonyms of the subjects or objects in one set, and use each cluster to represent the new subject or object. In this way, a canonicalized directed hypergraph dataset is obtained. Since it contains about 15 K entities, we call it ReVerb15K. The treated subject entities represent head hyperlinks, and the treated object entities represent the corresponding tails; the direction is from head to tail.

The specific size of datasets are shown in the Table 3.

Table 3. Statistics of the hypergraph datasets used in the experiments.

Datasets	WikiPeople	JF17K	ReVerb15K
Direction	undirected	undirected	directed
Number of Entities	12,270	11,541	14,798
Number of Relations	66	104	382

4.1.2. Metrics And Parameters

We test the effectiveness of the model in two parts. One is the Tucker-decomposition-based model for predicting the undirected hyperlinks, the other is the whole framework for predicting the directed hyperlinks. The total hyperlinks in datasets are divided into three parts: 20% for training, 10% for validation, and 70% for testing. We evaluate the link prediction performance via two standard metrics: MRR and Hits@k (k is top ranking). MRR is the mean of the inverse of rankings over all testing facts, while Hits@k measures the proportion of top k rankings. The aim of the training is to achieve high MRR and Hits@k.

The reported results are given for the best set of hyper-parameters evaluated on the validation set for each model, after grid search on the following values: embedding size $\in \{15, 20, 25, 30, 35\}$, learning rate $\in \{1, 0.6, 0.06, 0.006\}$, and TR-ranks $\in \{5, 10, 20, 30, 40\}$, with TR-ranks the size of the tensor decomposed by TR decomposition.

4.1.3. Baselines

We compare TF-DHP with the following n-ary hyperlink prediction baselines:

- RAE [8]: RAE is a translational distance model which considers the possibility of common occurrence between entities in n-order relations, establishes a correlation model through MLP, and reflects it in the scoring function.
- NaLP [11]: NaLP is a neural network model that achieves the state-of-the-art n-ary hypergraph link prediction performance.
- HGNN [12]: This is a general hypergraph neural network framework for data representation learning based on hypergraph convolution operation, which can incorporate multi-modal data and complicated data correlations. We use $maxmin_+$ as a scoring layer and a direction scoring layer [9] for directed hyperlink prediction with HGNN.
- HyperGCN [13]: This is a new method of training a GCN on hypergraph using tools from spectral theory of hypergraphs. Since it is not directly proposed for hyperlink prediction, we use the same scoring layers as used on HGNN.

- NHP-U-mean and NHP-U-maxmin [9]: These two methods are both based on the GCN layer. NHP-U-mean uses *mean* as the scoring layer while NHP-U-maxmin uses $maxmin_+$ as the scoring layer to predict hyperlinks. These two methods are proposed for undirected hyperlink prediction.
- NHP-D-mean and NHP-D-maxmin [9]: These two methods use a direction scoring layer on NHP-U-mean and NHP-U-maxmin to predict directed hyperlinks.

4.2. Experiment on Undirected Hypergraphs

Tables 4 and 5 show the undirected hyperlink prediction results on two datasets. The highest scores are set in bold. As shown in the tables, we can find out that our proposed TF-DHP can achieve optimal results under various measurement standards, consistently. For both datasets, graph neural networks NHP combining the *mean* or *maxmin* scoring functions cannot have comparable performances in link prediction problems. For example, on WikiPeople, compared with our proposed model, TF-DHP, the MRR of the first four methods is only about a third, and Hits@10 is about a half. The large improvement of TF-DHP can strongly confirm that scoring functions such as *mean* or *maxmin* largely ignore the influence of the representation of each entity in the hyperlink on the predicted results, which also reflects the advantage of Tucker-decomposition-based model taking every entity embedding into the computation.

Table 4. Undirected hyperlink prediction results on WikiPeople dataset.

Model	MRR	Hits@10	Hits@3	Hits@1
HGNN	0.132	0.285	0.152	0.117
HyperGCN	0.137	0.289	0.158	0.115
NHP-U-mean	0.122	0.283	0.147	0.119
NHP-U-maxmin	0.143	0.302	0.144	0.139
RAE	0.153	0.273	0.152	0.146
NaLP	0.332	0.537	0.403	0.334
TF-DHP	**0.362**	**0.574**	**0.440**	**0.368**

Table 5. Undirected hyperlink prediction results on JF15K dataset.

Model	MRR	Hits@10	Hits@3	Hits@1
HGNN	0.649	0.722	0.640	0.526
HyperGCN	0.654	0.743	0.652	0.538
NHP-U-mean	0.632	0.710	0.639	0.509
NHP-U-maxmin	0.686	0.783	0.670	0.573
RAE	0.707	0.837	0.751	0.629
NaLP	0.714	0.805	0.737	0.673
TF-DHP	**0.751**	**0.873**	**0.786**	**0.686**

As for the translational distance model RAE, although RAE achieves slightly better results than the four methods, its results are still unsatisfying. On WikiPeople, TF-DHP improves MRR by 0.21 and Hits@1 by 0.15, which is a considerable improvement. The main reason for the unsatisfying performance of RAE is the restriction on relations of the translational distance model. Such restriction does not exist in the Tucker-decomposition-based model. Tucker decomposition can accurately represent any ground truth over a set of entities and relations by its full expressiveness [15].

The performance of NaLP is much better than the aforementioned methods due to the enormous amount of model parameters. It uses a neural network to greatly reduce the restriction on relations existing in the translational distance model. However, a large number of parameters makes it easy to over-fit, especially when training datasets are not big enough. According to the network structure and scoring function of NaLP, the model complexity of NaLP is $O(n_e d_e + n n_r d_r)$, with n_e and d_e representing the number and dimension of entities, respectively. n is the number of entities in one relation. n_r and d_r stand for the number and the dimension of relations, respectively. However, the model complexity of the first stage of TF-DHP is only $O(n_e d_e + n_r d_r + n d_{max}^3)$, where d_{max} is the maximum dimension of the third-order tensors in TR decomposition. Since the number of relations is much larger than the dimension of the decomposed tensor in hypergraphs, the model complexity of NaLP is apparently larger than TF-DHP. As shown in Figure 3, with the training epoch growing, NaLP requires more training epochs than TF-DHP to achieve the optimal result. Moreover, because too many NaLP parameters lead to an over-fitting issue, the results decrease when the epoch is larger than 100. However, due to relatively few parameters, the results of TF-DHP are relatively stable after reaching the optimal result during training.

4.3. Experiment on Directed Hypergraphs

Table 6 shows the results of several directed hyperlink prediction models. The highest scores are set in bold. To the best of our knowledge, there are few models dealing with the hyperlink prediction problem in directed hypergraphs. As shown in Table 6, TF-DHP obtains considerable improvement compared with other methods. For example, for the best baseline NHP-D-maxmin, TF-DHP improves MRR by 0.056 and Hits@10 by 0.026.

Table 6. Directed hyperlink prediction results of Reverb15K dataset.

Model	MRR	Hits@10	Hits@3	Hits@1
HGNN	0.276	0.422	0.336	0.226
HyperGCN	0.316	0.443	0.347	0.238
NHP-D-mean	0.288	0.435	0.352	0.219
NHP-D-maxmin	0.442	0.560	0.438	0.348
TF-DHP	**0.498**	**0.586**	**0.474**	**0.353**

We believe that there are two main reasons for the better prediction performance of TF-DHP on directed hypergraphs. First, when testing the directed hypergraph prediction model, we put the weighted average of scores computed in two stages of the model as the final score, which means we regard an entity set as positive only if there exists a directed hyperlink among the entities set with the direction also being correct. So, the accuracy of the first stage of the model will inevitably affect the performance of the whole model. Second, the NHP-D-maxmin and other methods in Table 6 use the average value of entities' embedding vectors to represent the embedding of the hyperlink and consider the product of embedding vectors of *head* and *tail* parts of the hyperlink as a scoring function. As mentioned above, these methods ignore the influence of each entity embedding on the direction of the hyperlink and the relationship between an entity and its adjacent entities. The improvement of experimental results proves that considering the representation information of each entity separately and the information of the adjacent entities (from forward to backward) can improve the accuracy of directed hypergraph prediction.

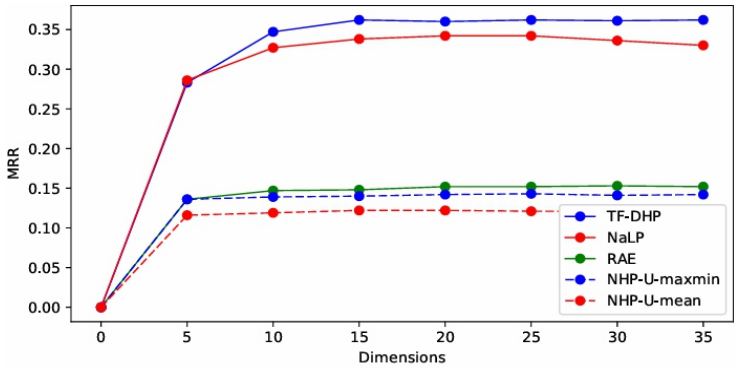
4.4. Parameter Analysis

Embedding size is a significant factor in hyperlink prediction models, determining the performance of the model to a large extent. Hence, we will analyze the results obtained by the model in different embedding sizes to investigate its impact.

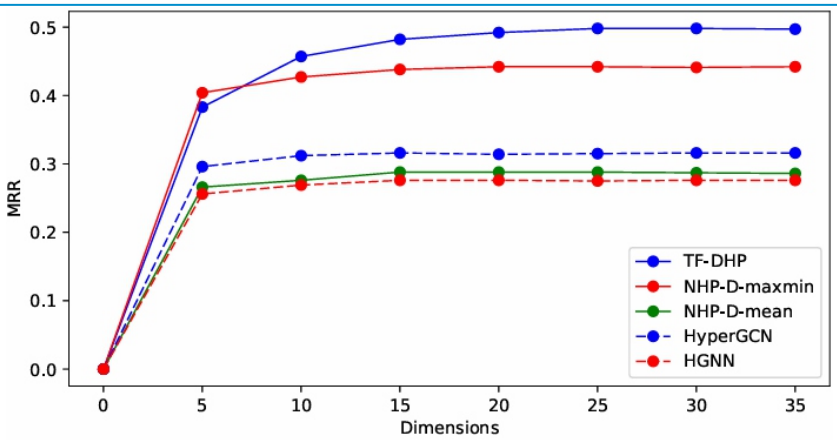
First, according to Figure 5a, TF-DHP outperforms other methods on each embedding size. The MRR of TF-DHP increases sharply with the early stage of increasing the embedding size and becomes smooth after the embedding size increases to 15. The MRR of NaLP is almost identical to TF-DHP's from the start; however, due to a large number of parameters, it cannot reamain smooth like TF-DHP when the embedding size increases. After the embedding size increases to a certain extent, NaLP's MRR will decrease. For other methods, the change in embedding size has less influence on the experimental results due to their smaller number of parameters.

Figure 5b shows the impacts of embedding size on directed hyperlink prediction. The same as undirected hyperlink prediction, TF-DHP always outperforms other methods. As BiLSTM is added, the optimal embedding size of the model increases to 25, after which the increase in MRR becomes smooth. As for other methods, the addition of the direction scoring function also increases the optimal number of parameters and shares the similar tendency as TF-DHP.

It proves the stability of TF-DHP on the choice of the dimension size. In addition the reasonable amount of parameters of TF-DHP allows it to be more stable, as other models' performances may decrease with the increasing dimensions, suffering from the over-fitting issue.



(a)



(**b**)

Figure 5. MRR over different embedding sizes of undirected hyperlink prediction models and directed hyperlink prediction models, evaluated on WikiPeople and Reverb15k: (**a**) undirected hyperlink prediction models; (**b**) directed hyperlink prediction models.

4.5. Approximate Training Time Comparison

On the two undirected datasets WikiPeople and JF15K, TF-DHP takes around 45 min of training time, while NaLP and RAE take around 3 h and 1 h, respectively. On the directed dataset Reverb15K, TF-DHP takes around 1 h of training time, while NHP-D-maxmin and NHP-D-mean take around 15 min each due to their oversimplified scoring function. All were run on a GeForce GTX 1080 super GPU machine.

4.6. Ablation Study

Since experiments on the directed hypergraph dataset have proved the effectiveness of the BiLSTM model, we designed an ablation study to prove the influence of TR decomposition in Tucker decomposition. We designed a variant on WikiPeople of TF-DHP which does not use TR decomposition on Tucker decomposition, and we call it *n-Tucker*. As shown in Figure 6, without TR decomposition, the computational complexity of the model greatly increases, which will result in an over-fitting issue. Similar but better than NaLP, n-Tucker reaches the optimal value of MRR and then gradually decreases due to the over-fitting issue. This kind of experiment not only proves the superiority of the Tucker decomposition-based model but also proves the necessity of the TR decomposition.

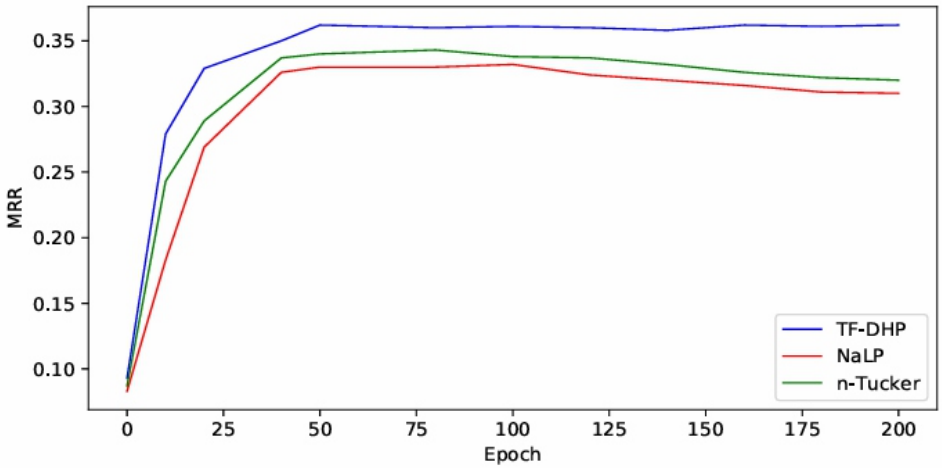


Figure 6. MRR under different training epochs of undirected hyperlink prediction models. Evaluated on WikiPeople.

5. Conclusions and Future Work

In this paper, we introduce TF-DHP, a novel model for hyperlink prediction for both undirected and directed hypergraphs. We use a tensor-decomposition-based method to handle the undirected part and add a BiLSTM model to predict the direction of the hyperlink. Our model TF-DHP is a pipelined model, which is flexible to deal with not only directed hypergraphs but also undirected hypergraphs. The experimental results verify the advantages of TF-DHP in both settings across multiple datasets.

In the future, we plan to further look into heterogeneous hypergraphs where there are multiple types of high-order relations, such as inclusion relations and produce relations, and to see how directed hypergraphs can be used on reaction prediction in chemical or biological domains.

Author Contributions: Conceptualization, G.X. and J.L.; methodology, G.X.; software, G.X. and J.L.; validation, G.X., J.L. and X.Z. (Xiang Zhao); formal analysis, G.X.; investigation, G.X., J.L., Z.T. and X.Z. (Xiaonan Zhang); data curation, G.X.; writing—original draft preparation, G.X.; writing—review and editing, G.X., J.L. and X.Z. (Xiang Zhao); visualization, G.X. All authors have read and agreed to the published version of the manuscript.

Funding: This work was partially supported by NSFC under grants Nos. 61872446, 61902417, and The Science and Technology Innovation Program of Hunan Province under grant No. 2020RC4046.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: Not applicable.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Zhang, M.; Chen, Y. Link Prediction Based on Graph Neural Networks. In Proceedings of the Advances in Neural Information Processing Systems 31: Annual Conference on Neural Information Processing Systems 2018 (NeurIPS 2018), Montréal, QC, Canada, 3–8 December 2018; pp. 5171–5181.

2. Ning, X.; Shen, L.; Li, L. Predicting High-Order Directional Drug-Drug Interaction Relations. In Proceedings of the 2017 IEEE International Conference on Healthcare Informatics, ICHI 2017, Park City, UT, USA, 23–26 August 2017; IEEE Computer Society: Washington, DC, USA, 2017; pp. 556–561.

3. Jin, T.; Wu, Q.; Ou, X.; Yu, J. Community detection and co-author recommendation in co-author networks. Int. J. Mach. Learn. Cybern. 2021, 12, 597–609. [CrossRef]

4. Zhang, Z.; Liu, C. Hypergraph model of social tagging networks. arXiv 2010, arXiv:1003.1931.

5. Bollobás, B.; Daykin, D.E.; Erdös, P. Sets of Independent edges of a hypergraph. Q. J. Math. 1976, 27, 25–32. [CrossRef] 6. Li, D.; Xu, Z.; Li, S.; Sun, X. Link prediction in social networks based on hypergraph. In Proceedings of the 22nd International Conference on World Wide Web, Rio de Janeiro, Brazil, 13–17 May 2013.

7. Wen, J.; Li, J.; Mao, Y.; Chen, S.; Zhang, R. On the Representation and Embedding of Knowledge Bases beyond Binary Relations. In Proceedings of the Twenty-Fifth International Joint Conference on Artificial Intelligence, IJCAI 2016, New York, NY, USA, 9–15 July 2016; Kambhampati, S., Ed.; IJCAI/AAAI Press: Palo Alto, CA, USA, 2016; pp. 1300–1307.

8. Zhang, R.; Li, J.; Mei, J.; Mao, Y. Scalable Instance Reconstruction in Knowledge Bases via Relatedness Affiliated Embedding. In Proceedings of the 2018 World Wide Web Conference on World Wide Web, WWW 2018, Lyon, France, 23–27 April 2018; Champin, P., Gandon, F., Lalmas, M., Ipeirotis, P.G., Eds.; ACM: New York, NY, USA, 2018; pp. 1185–1194.

9. Yadati, N.; Nitin, V.; Nimishakavi, M.; Yadav, P.; Louis, A.; Talukdar, P.P. NHP: Neural Hypergraph Link Prediction. In Proceedings of the CIKM '20: The 29th ACM International Conference on Information and Knowledge Management, Virtual Event, Ireland, 19–23 October 2020; ACM: New York, NY, USA, 2020; pp. 1705–1714.

10. Wang, Z.; Zhang, J.; Feng, J.; Chen, Z. Knowledge Graph Embedding by Translating on Hyperplanes. In Proceedings of the AAAI, Quebec City, QC, Canada, 27–31 July 2014.

11. Guan, S.; Jin, X.; Wang, Y.; Cheng, X. Link Prediction on N-ary Relational Data. In Proceedings of the The World Wide Web Conference (WWW 2019), San Francisco, CA, USA, 13–17 May 2019; Liu, L., White, R.W., Mantrach, A., Silvestri, F., McAuley, J.J., Baeza-Yates, R., Zia, L., Eds.; ACM: New York, NY, USA, 2019; pp. 583–593.

12. Feng,Y.; You, H.; Zhang, Z.; Ji, R.; Gao, Y. Hypergraph Neural Networks. In Proceedings of the the Thirty-Third AAAI Conference on Artificial Intelligence (AAAI 2019), The Thirty-First Innovative Applications of Artificial Intelligence Conference (IAAI 2019), The Ninth AAAI Symposium onEducational Advances in Artificial Intelligence (EAAI 2019), Honolulu, HI, USA, 27 January–1 February 2019; AAAI Press: Palo Alto, CA, USA, 2019; pp. 3558–3565.

13. Yadati, N.; Nimishakavi, M.; Yadav, P.; Nitin, V.; Louis, A.; Talukdar, P.P. HyperGCN: A New Method For Training Graph Convolutional Networks on Hypergraphs. In Proceedings of the Advances in Neural Information Processing Systems 32: Annual Conference on Neural Information Processing Systems 2019 (NeurIPS 2019), Vancouver, BC, Canada, 8–14 December 2019; pp. 1509–1520.

14. Kipf, T.N.; Welling, M. Semi-Supervised Classification with Graph Convolutional Networks. In Proceedings of the 5th International Conference on Learning Representations (ICLR 2017), Toulon, France, 24–26 April 2017.

15. Balazevic, I.; Allen, C.; Hospedales, T.M. TuckER: Tensor Factorization for Knowledge Graph Completion. In Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP 2019), Hong Kong, China, 3–7 November 2019; Inui, K., Jiang, J., Ng, V., Wan, X., Eds.; Association for Computational Linguistics: Stroudsburg, PA, USA, 2019; pp. 5184–5193.

16. Zhao, Q.; Zhou, G.; Xie, S.; Zhang, L.; Cichocki, A. Tensor Ring Decomposition. arXiv 2016, arXiv:1606.05535.

17. Tamburini, F. A BiLSTM-CRF PoS-tagger for Italian tweets using morphological information. In Proceedings of the Third Italian Conference on Computational Linguistics (CLiC-it 2016) & Fifth Evaluation Campaign of Natural Language Processing and Speech Tools for Italian. Final Workshop (EVALITA 2016), Napoli, Italy, 5–7 December 2016; Volume 1749.

18. Nickel, M.; Tresp, V.; Kriegel, H. A Three-Way Model for Collective Learning on Multi-Relational Data. In Proceedings of the 28th International Conference on Machine Learning, ICML 2011, Bellevue, WA, USA, 28 June–2 July 2011; Getoor, L., Scheffer, T., Eds.; Omnipress: Madison, WI, USA, 2011; pp. 809–816.

19. Yang, B.; Yih, W.; He, X.; Gao, J.; Deng, L. Embedding Entities and Relations for Learning and Inference in Knowledge Bases. In Proceedings of the 3rd International Conference on Learning Representations (ICLR 2015), San Diego, CA, USA, 7–9 May 2015.

20. Trouillon, T.; Welbl, J.; Riedel, S.; Gaussier, É.; Bouchard, G. Complex Embeddings for Simple Link Prediction. In Proceedings of the 33nd International Conference on Machine Learning (ICML 2016), New York, NY, USA, 19–24 June 2016; Volume 48, pp. 2071–2080.

21. Kazemi, S.M.; Poole, D. SimplE Embedding for Link Prediction in Knowledge Graphs. In Proceedings of the Advances in Neural Information Processing Systems 31: Annual Conference on Neural Information Processing Systems 2018 (NeurIPS 2018), Montréal, QC, Canada, 3–8 December 2018; pp. 4289–4300.

22. Dettmers, T.; Minervini, P.; Stenetorp, P.; Riedel, S. Convolutional 2D Knowledge Graph Embeddings. In Proceedings of the Thirty-Second AAAI Conference on Artificial Intelligence, (AAAI-18), the 30th innovative Applications of Artificial Intelligence (IAAI-18), and the 8th AAAI Symposium on Educational Advances in Artificial Intelligence (EAAI-18), New Orleans, LA, USA, 2–7 February 2018; McIlraith, S.A., Weinberger, K.Q., Eds.; AAAI Press: Palo Alto, CA, USA, 2018; pp. 1811–1818.

23. Balazevic, I.; Allen, C.; Hospedales, T.M. Hypernetwork Knowledge Graph Embeddings. In Proceedings of the Artificial Neural Networks and Machine Learning—ICANN 2019—28th International Conference on Artificial Neural Networks, Munich, Germany, 17–19 September 2019; Proceedings—Workshop and Special Sessions; Lecture Notes in Computer Science; Tetko, I.V., Kurková, V., Karpov, P., Theis, F.J., Eds.; Springer: Berlin/Heidelberg, Germany, 2019; Volume 11731, pp. 553–565.

24. Liu, W.; Lu, L. Link prediction based on local random walk. EPL 2010, 89, 58007. [CrossRef]

25. Berahmand, K.; Nasiri, E.; Forouzandeh, S.; Li, Y. A Preference Random Walk Algorithm for Link Prediction through Mutual Influence Nodes in Complex Networks. arXiv 2021, arXiv:2105.09494.

26. Nasiri, E.S.; Berahmand, K.; Li, Y. A new link prediction in multiplex networks using topologically biased random walks. Chaos Solitons Fractals 2021, 151, 111230. [CrossRef]

27. Zhang, R.; Zou, Y.; Ma, J. Hyper-SAGNN: A self-attention based graph neural network for hypergraphs. In Proceedings of the 8th International Conference on Learning Representations, ICLR 2020, Addis Ababa, Ethiopia, 26–30 April 2020.

28. Vashishth, S.; Jain, P.; Talukdar, P.P. CESI: Canonicalizing Open Knowledge Bases using Embeddings and Side Information. In Proceedings of the 2018 World Wide Web Conference on World Wide Web, WWW 2018, Lyon, France, 23–27 April 2018; Champin, P., Gandon, F., Lalmas, M., Ipeirotis, P.G., Eds.; ACM: New York, NY, USA, 2018; pp. 1317–1327.

29. Fader, A.; Soderland, S.; Etzioni, O. Identifying Relations for Open Information Extraction. In Proceedings of the 2011 Conference on Empirical Methods in Natural Language Processing, EMNLP 2011, Edinburgh, UK, 27–31 July 2011; A meeting of SIGDAT, a Special Interest Group of the ACL; ACL: Stroudsburg, PA, USA, 2011; pp. 1535–1545.

30. Gabrilovich, E.; Ringgaard, M.; Subramanya, A. FACC1: Freebase Annotation of ClueWeb Corpora, Version 1 (Release date 2013-06-26, Format Version 1, Correction Level 0). 2013.

31. Callan, J.; Hoy, M.; Yoo, C.; Zhao, L. Clueweb09 Data Set 2009

Representing Hierarchical Structured Data Using Cone Embedding

Daisuke Takehara and Kei Kobayashi

ALBERTInc., Shinjuku Front Tower 15F 2-21-1, Kita-Shinjuku, Shinjuku-ku, Tokyo 169-0074, Japan Department of Mathematics, Faculty of Science and Technology, Keio University, 3-14-1 Hiyoshi, Kohoku-ku, Correspondence:

ABSTRACT

Extracting hierarchical structure in graph data is becoming an important problem in fields such as natural language processing and developmental biology. Hierarchical structures can be extracted by embedding methods in non-Euclidean spaces, such as Poincaré embedding and Lorentz embedding, and it is now possible to learn efficient embedding by taking advantage of the structure of these spaces. In this study, we propose embedding into another type of metric space called a metric cone by learning an only one-dimensional coordinate variable added to the original vector space or a pre-trained embedding space. This allows for the extraction of hierarchical information while maintaining the properties of the pre-trained embedding. The metric cone is a one-dimensional extension of the original metric space and has the advantage that the curvature of the space can be easily adjusted by a parameter even when the coordinates of the original space are fixed. Through an extensive empirical evaluation we have corroborated the effectiveness of the proposed cone embedding model. In the case of randomly generated trees, cone embedding demonstrated superior performance in extracting hierarchical structures compared to existing techniques, particularly in high-dimensional settings. For WordNet embeddings, cone embedding exhibited a noteworthy correlation between the extracted hierarchical structures and human evaluation outcomes.

Keywords: *graph embedding; non-Euclidean space; WordNet*

Introduction

In recent years, machine learning methods for graph data have been an important topic, because graphs are suitable for representing the relation between multiple objects, such as social networks [1,2], links embedded in web pages [3], cells' interactions [4], and more. In particular, methods for extracting hierarchical structures from graph data are needed in fields such as cell engineering and natural language processing. Considering the structure of knowledge behind language is important for natural language processing tasks in general. The hierarchical structure of words provides useful information for improving the accuracy of question answering and semantic search [5,6]. In the field of developmental biology, various methods have been proposed for analyzing single-cell RNA sequence (scRNAseq) data to reveal the process by which an undeveloped cell develops into a cell with specific features [7]. Since scRNAseq data itself does not have a hierarchical structure, the hierarchical structure must be extracted from the data or from a graph constructed using the data. The methodforextracting hierarchical structures must have some scalability when it is applied to data sets with a large size and high dimensions such as scRNAseq data. The most common method for extracting the structure of a graph is to learn the embedding vector of nodes. Methods for learning node embeddings can be classified into two types: (1)

semi-supervised learning based on GNN [8–10] and (2) unsupervised learning [11] (based on random-walk [12], matrix factorization [13], and probabilistic methods [14], etc.). Graph neural networks (GNNs) are a type of neural network designed to operate on graph-structured data, allowing them to model complex relationships between entities, and capture both local and global information in the graph. This is achieved through the use of message passing mechanisms, which enable nodes to exchange information with their neighbors and aggregate that information into a new representation. Although it is possible to solve tasks that require hierarchical structure information using only GNNs, there are many advantages to using embedded representations, such as the expected reduction in computational complexity if the hierarchical structure is extracted in advance for embedding. On the other hand, the graph embedding converts each graph into a vector representing features of the graph and such vector representation can be tuned for solving individual tasks, which reduces the overall computational complexity. In this paper, we propose a novel graph embedding method for extracting its hierarchical structure from an undirected graph. There have been many graph embedding methods for extracting the hierarchical structure of a graph utilizing a hyperbolic space [15,16], such as Poincaré embedding [17–20], Lorentz embedding [21], and embedding in a hyperbolic entailment cone [22]. These methods use similar loss functions but with different metrics of the space in which graphs are embedded. Non-Euclidean spaces with non-zero curvature can learn embedding efficiently by adjusting their curvature to the hierarchically structured data. In particular, a Poincaré ball is a space of a negative constant curvature, which is characterized by the fact that the length of the circumference exponentially increases in the order of the radius when centered at the origin. An efficient embedding of tree-structured data utilizing this feature has also been proposed [23]. The Lorentz model of a hyperbolic space can explicitly describe geodesics and the accuracy of distance calculation becomes stable in the optimization [21]. The metric cone used as the embedding space in this study is a space defined as a one-dimensional extension of a base metric space. The base metric space can be not only a vector space but for any geodesic metric space such as Riemannian manifolds and metric graphs. The dimensions of the metric cone are only one dimension higher than the original space. It is known that the curvature of this space can be varied and a method of changing the structure of the data space for analysis has also been proposed [24]. The definition and details of the metric cone will be explained in Section 2.3. In this paper, we propose the use of the metric cone as an embedding method for hierarchical graphs. Thanks to the properties of metric cones, the proposed method has the following five advantageous features compared to existing methods. First, it optimizes an only one-dimensional coordinate corresponding to "the height of the metric cone" (a one-dimensional parameter added to the base space) as an indicator of hierarchy. Therefore, a significant reduction in computational complexity can be expected compared to optimizing all variables. Secondly, it can be applied to any pre-trained embeddings using a geodesic metric space including the Poincaré ball and the Lorentz model. When extracting hierarchical information for another purpose from an embedding already learned by other embedding methods, the

extraction of hierarchical structure can be accomplished by learning only one additional coordinate variable. Due to this scalability, the proposed method can be combined with various existing embedding methods to achieve hierarchical extraction with a variety of features. Thirdly, the curvature of embedding space varies monotonically with , a parameter in the distance function of embedding space, and therefore can be tuned by it. As explained in Section 3.2, parameter corresponds to the generatrix of the metric cone and this fact provides an intuitive explanation for the monotonically decreasing curvature of the embedding space as the parameter is increased; while there have been some methods for tuning the curvature of some graph embedding spaces [25,26], the metric cone allows the curvature of the space to be tuned by changing while keeping the coordinates of the original space fixed. Therefore, when adjusting the curvature of the embedding space to match the training data, only one-dimensional parameters need to be learned. As shown in the experiments, it is suitable to embed data with a smaller curvature in higher dimensions. Thus, it is important to adjust curvature depending on the dimension of the destination space and the structure of the data to be embedded. Fourthly, the uniqueness of the embedding is guaranteed when optimizing the loss function. When performing graph embedding in a space where isometric transformations exist, there is the problem of unstable learning due to the existence of multiple embeddings such that the distance from the origin of each point can be different, even though the distances between all points are identical. Usually, the distance from the origin is used as the height of the hierarchy, resulting in multiple solutions with different hierarchical structures. On the other hand, since there is no isometric mapping for a sufficiently large number of points in a metric cone as proven in Section 3.1, it is theoretically guaranteed that the embedding is unique and the learning is stable. Lastly, we can reduce the amount of computation for the parts other than preprocessing, regardless of the dimension. In addition, because the embedding in the original Euclidean space is preserved, it can be used as an input to the neural network and can be easily applied to other tasks. The subsequent sections of this paper are organized as follows. First, in Section 2, wepropose the method of graph embedding in a metric cone, with the introduction of (1) graph embedding in non-Euclidean spaces, and (2) the definition and properties of cones. In Section 3, theoretical arguments ensure the validity of the proposed method. First, weprove that the identifiability of the graph embedding, which does not hold for existing methods, holds for the cone embedding. Next, we show that the curvature of the metric cone varies monotonically with the parameter . In Section 4, we present experimental results using some real and artificial graph data, followed by a conclusion and future perspectives in Section 5.

2. Methods

2.1. Problem Settings

From this point onward, the set of edges in an undirected graph G is denoted by E, the set of vertices by V, and the embedded space by X. Then, our target is finding an embedding : V Xandafunctionh : X Rsuchthath((v))representsthehierarchy of v V.Function h canusually be expressed simply as a coordinate value of X. Note that, since G is an undirected graph, the problem is ill-posed if there are no

assumptions about the relationship between the structure of the graph and the hierarchy of vertices. As in existing works, we implicitly assume that the branching of the graph is like that of a rooted tree, i.e., the higher the hierarchy, the smaller the number of vertices, and the lower the hierarchy, the more vertices.

2.2. Graph Embedding in Non-Euclidean Spaces

Out learning steps are similar to Poincaré embedding. We learn the embedding of a graph G bymaximizing the following objective function:

$$L = \sum_{(u,v)\in E} \log \frac{\exp(-d(u,v))}{\sum_{v'\in N^c(u)} \exp(-d(u,v'))}, \tag{1}$$

where $N^c(u) : \{v' \in V|(u,v') \notin E\}$ denotes the set of points not adjacent to node u (including u itself) and d denotes the distance function of the embedded space. Here the embedded space becomes a Poincaré sphere for the Poincaré embedding and a metric cone for the proposed method. This objective function is a negative sampling approximation of a model in which the similarity is -1 times the distance and the probability of the existence of each edge is represented by a SoftMax function on the similarity.

The maximization of the objective function is done by stochastic gradient descent on Riemannian manifolds (Riemannian SGD). The stochastic gradient descent over Euclidean space updates the parameters as follows:

$$u \leftarrow u - \eta \nabla_u L(u), \tag{2}$$

where η is the learning rate. However, in non-Euclidean, the sum of vectors is not defined and $\nabla_u L(u)$ is the point of the tangent space $T_u X$ of u; hence, SGD cannot be applied. Therefore, we update the parameters by using an exponential map instead of the sum:

$$u \leftarrow \exp_u(-\eta \nabla_u^R L(u)). \tag{3}$$

With the metric tensor of the embedding space as $g_u(u \in V)$, the gradient on the Riemannian manifold $\nabla_u^R L(u)$ is the scaled gradient in Euclidean space:

$$\nabla_u^R L(u) = g_u^{-1} \nabla_u L(u). \tag{4}$$

2.3. The Metric Cone

The metric cone is similar to ordinary cones (e.g., circle cones) in the sense that it is defined as a collection of line segments connecting an apex point to a given set. However, the metric cone has a notable property such that every point in the original set is embedded

at an equal distance from the apex point and this is a desirable property for hierarchical structure extraction. The metric cone has been studied as an analogy to the length metric spaces of the tangent cone for differential manifolds with singularities. Length metric space is a metric space where the distance between any two points is equal to the shortest curve length connecting them. Length metric space includes Euclidean spaces, normed vector spaces, manifolds (e.g., Poincaré ball; sphere), metric graphs, and many other metric spaces. Assume the original space Z is a length metric space, then the

metric cone generated by Z is X := Z [0,1] / Z 0 with a distance function determined as follows:

$$\tilde{d}_\beta((x,s),(y,t)) = \beta\sqrt{t^2+s^2-2ts\cos(\pi\min(d_Z(x,y)/\beta,1))} \quad (5)$$

where $\beta > 0$ is a hyperparameter corresponding to the length of the conical generatrix. Note that the metric cone itself also becomes a length metric space and it embeds the original space (i.e., the space is one dimension larger than the original space). The distance in the metric cone corresponds to the length of the shortest curve on the circle section (blue line segment(s) in the right two subfigures in Figure 1) whose bottom circumference is the distance of the original space Z and whose radius is β.

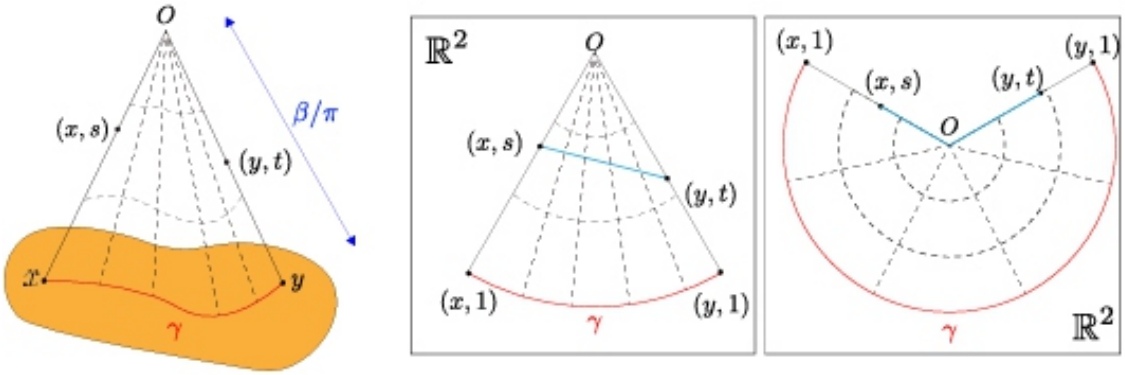


Figure 1. The left figure depicts a conceptual image of an original space and its metric cone. A circle section to compute the distance in the metric cone is depicted in the middle figure (when the apex angle $< \pi$) and the right figure (when the apex angle $\geq \pi$).

When the curvature is measured in the sense of CAT(k) property, a curvature measure for general length metric spaces, the curvature value k can be controlled by β. Other properties of the metric cone are examined in [27,28]. Because the metric cone can change the curvature of the space by changing parameter β, its usefulness has been reported in an analysis using the structure of the data space [24].

The metric $\tilde{g}$ of a metric cone is obtained by calculating the two-time derivative of the distance as follows (see Appendix A for more details):

$$\tilde{g}_{(x,s)} = \begin{pmatrix} \pi^2 s^2 g_x & 0 \\ 0^\top & \beta^2 \end{pmatrix}, \quad (6)$$

where g_x represents the metric of Z at x. Combining this metric and the argument in Section 3.1, the algorithm of cone embedding can be described as Algorithm 1.

Algorithm 1 Learn the cone embedding $\{(u,s)\}$

Input: graph $G = (V,E)$, cone's hyperparameter β , learning rate η, and the pre-trained embedding $\{x\}$ in original space Z

Output: the cone embedding $\{(u,s)\}$

1: calculate the distance matrix $D = (d_{ij})$, $d_{ij} = d_Z(x_i, x_j)$

2: minimize the softmax loss function:
(calculate efficiently by referencing the distance matrix D)

$$L = \sum_{((u,s),(v,t))\in E} \log \frac{\exp^{-\bar{d}_\beta((u,s),(v,t))}}{\left(\sum_{(v',t')\in N^c((u,s))} \exp\left(-\bar{d}_\beta((u,s),(v',t'))\right)\right)} \tag{7}$$

via Riemannian stochastic gradient descent:

$$(x,s) \leftarrow \mathrm{proj}\Big((x,s) - \eta \bar{g}^{-1}\nabla L\Big)$$

The loss function (7) is defined to be smaller if the distance between nodes sharing an edge becomes smaller (by the numerator in the log) and the distance between nodes without an edge becomes larger (by the denominator in the log). Note that the distance for a metric cone is used here. Computation of the denominator can be reduced by random sampling of nodes for which no edges exist. Furthermore, the projection normalizes the embedding along the gradient so that it does not jump out of the metric cone when it is updated.

Instead of the exponential map of the metric cone, we use the first-order approximation using $\mathrm{proj}(x,s)$:

$$\mathrm{proj}(x,s) = \begin{cases} (x,s) & \text{if} \quad \epsilon < s < 1-\epsilon, \\ (x,1-\epsilon) & \text{if} \quad s \geq 1-\epsilon, \\ (x,\epsilon) & \text{if} \quad s \leq \epsilon. \end{cases} \tag{8}$$

2.4. Score Function of Hierarchy

The Poincaré embedding defines an index in [17], which is aimed to be an indicator of the hierarchical structure and depends on the distance from the origin:

$$\mathrm{score}(u,v) = -\alpha(\|v\| - \|u\|)d(u,v) \tag{9}$$

This score function is penalized by the part after α, so, if v is closer to the origin than u, then it is easier to obtain larger values. In other words, v is higher in the hierarchy than u (i.e., "u is a v" relationship holds). However, it is not appropriate to use this indicator for the Poincaré embedding. This model learns the embedding by maximizing (1), where

$$d(x,y) := \mathrm{arcosh}\left(1 + \frac{\|x-y\|^2}{(1-\|x\|^2)(1-\|y\|^2)}\right). \tag{10}$$

This loss function only depends on the distance between the two embeddings. However, an isometric transformation in the Poincaré ball exists, known as the Möbius transformation [29]. Möbius transformation is defined as a map $f : \mathbb{B}^n(\text{open unit ball}) \to \mathbb{B}^n$, which can be written as a product of the inversions of $\bar{\mathbb{R}}^n (:= \mathbb{R}^n \cup \{\infty\})$ through a sphere S that preserves $\mathbb{B}^n$.

In contrast to the Poincaré ball, the isometric transformation on the metric cone does not exist when the coordinate in the original space is fixed (we prove this property in Appendix C). When we embed a graph into a metric cone, we define an indicator of the hierarchical structure by replacing the norm with a coordinate corresponding to the height of the cone (a one-dimensional parameter added to the original space):

$$\mathrm{score}((\mathbf{u},\mathrm{s}),(\mathbf{v},\mathrm{t})) = -\alpha(s-t)d(\mathbf{u},\mathbf{v}). \tag{11}$$

A point closer to the top of the cone is higher in the hierarchy, which is natural for representing hierarchical structure.

2.5. Using Pre-Trained Model for Computational Efficiency and Adaptivity for Adding Hierarchical Information

Consider a situation where we already have a trained graph embedding on a Euclidean space (e.g., LINE [14]) and we try to learn the embedding in a metric cone of Euclidean space to extract information about the hierarchical structure. In this case, we can reduce the computational cost by fixing the coordinates corresponding to the original Euclidean space and learn only the one-dimensional parameters corresponding to heights in a metric cone added to the original space because the metric cone is one dimension larger than the original space. The distance between each embedding in the original space is calculated beforehand, since no updates are made by learning except for the 1D parameter to be added. By referring to the pre-computed distances in the original space when calculating the distances between each embedding on the metric cone ($d_z(x, y)$ in Equation (5)) during training, we can reduce the amount of computation for the parts other than preprocessing, regardless of the dimension. In addition, because the embedding in the original Euclidean space is preserved, it can be used as an input to the neural network (when the task considers information about the hierarchy and the added one-dimensional parameters are also used as input) and can be easily applied to other tasks. However, other non-Euclidean embedding methods to extract hierarchical structures are not scalable because these methods cannot be applied directly to solve other tasks. For example, deep neural networks cannot use a non-Euclidean embedding as input because the sum of two vectors in the space and scalar product is not generally defined.

2.6. Comparison with Hierarchical Clustering

Although both cone embedding and hierarchical clustering aim to extract hierarchical structures, there is a clear difference between their problem settings. In hierarchical clustering, only leaves in a result tree (dendrogram) correspond to data points and other nodes correspond to created clusters. Thus, the problem setting differs significantly from cone embedding in which each node in a data graph corresponds to a pre-defined entity. As a result, hierarchical clustering cannot extract the hierarchy of nodes other than leaves while cone embedding can do. Moreover, the order of computational complexity is also different: hierarchical clustering requires $O(n^2)$, while cone embedding requires $O(|E|)$ ($|E|$: number of edges), making it suitable for extracting hierarchical structures in large graphs.

It has been also shown by [30] that the embedding of tree-structured (undirected) graph data can be done naturally in hyperbolic space, but graph data with hierarchical structure does not necessarily have a tree structure in general (e.g., there can be a cycle when a child node has two parents which have the same parent). Thus, the combination with hyperbolic embedding and hierarchical clustering may not be suitable in such cases. Cone embedding does not assume the tree structure and extracts the hierarchical structure by using the property that the closer to the origin O the shorter the distance between data points, so that embedding can be learned even in this situation.

3. Theory

In this section, we give theoretical proof as to why the spatial properties of the metric cone are suitable for extracting hierarchical structures.

3.1. Identifiability of the Heights in Cone Embedding

As mentioned above, for Poincaré embedding, there is an isometric transformation on the Poincaré ball and the heights of the learned hierarchy are not invariant to such transformation. Here, we show that such a phenomenon does not occur for the cone embedding, i.e., the heights of the hierarchy are (almost) uniquely determined from the distance between the embedded data points in a metric cone.

Let Z be an original embedding space (connected length metric space) and let X be a metric cone of Z with a parameter $\beta > 0$. We assume that each data point $z_i \in Z$ $(i = 1, \dots, n)$ has its specific "height" $t_i \in [0, 1]$ in the metric cone X. Our proposed method embeds data points into a metric cone based on the estimated distances $\bar{d}_\beta(x_i, x_j)$ $(i, j = 1, \dots, n)$ and tries to compute the heights $t_1, \dots, t_n$ as a measure of the hierarchy level. However, it is not evident whether these heights are identifiable only from the information of the original data points in Z and the distances $\bar{d}_\beta(x_i, x_j)$ $(i, j = 1, \dots, n)$ in the metric cone. The following theorem guarantees some identifiability. (A rigorous version of Theorem 1, including the precise meaning of "identifiable" in (a)–(c) and "general" in (b), is explained in Appendix C.)

Theorem 1. *(a) Let $n \geq 3$ and assume that $z_1, \dots, z_n$ are not all aligned on a geodesic in Z. Then, the heights $t_1, \dots, t_n$ are identifiable up to at most four candidates.*

(b) Let $n \geq 4$ and assume $z_1, \dots, z_n$ and $t_1, \dots, t_n$ take "general" positions and heights, respectively. Then, the heights $t_1, \dots, t_n$ are identifiable uniquely.

(c) If $d_Z(z_i, z_j) \geq \frac{\beta}{2}$ for all $i, j = 1, \dots, n$, $i \neq j$, then the heights $t_1, \dots, t_n$ are identifiable uniquely.

Theorem 1(a) indicates that the candidates of heights are finite and we can expect the algorithm to converge to one of them, except for a very special data distribution in the original space Z. Moreover, by (b), even the uniqueness can be proved under very mild conditions. The statement in (c) implies that the uniqueness holds for arbitrary data distributions when we set β sufficiently small.

Remark that the assumption of "general" positions in Theorem 1(b) is satisfied easily for most data distributions. For example, if both $z_1, \dots, z_n \in \mathbb{R}^d$ and $t_1, \dots, t_n \in [0, 1]$ are i.i.d. from a probability distribution whose density function exists with respect to the Lebesgue measure, then it is easy to see the assumption holds almost surely and therefore the uniqueness of the solution is guaranteed. Note that, for $n = 3$ under the same setting, there can be multiple solutions with a positive probability.

3.2. Variable Curvature

One of the essences of Poincaré embedding is that a negative curvature of the Poincaré sphere is suitable for embedding tree graphs. The curvature of a metric cone has a similar property, i.e., a metric cone has more negative curvature than the original space and, furthermore, the curvature can be controlled by hyperparameter β. We will verify these facts mathematically from two different aspects: (i) the scalar and the Ricci curvatures of a Riemannian manifold and (ii) the CAT(k) property of a length metric space.

First, assume the original space $\mathcal{M}$ is an n-dimensional Riemannian manifold with a metric g. Then the metric $\bar{g}$ of the corresponding metric cone with β can be defined except for the apex and it becomes as (6). Let $1, \dots, n$ be coordinate indices corresponding to $x \in \mathcal{M}$ and 0 be the index corresponding to $s \in [0, 1]$. The Ricci curvatures $\bar{R}_{ij}$ and the scalar curvature $\bar{R}$ at (x, s) become

$$\tilde{R}_{\alpha\gamma} = R_{\alpha\gamma} - \pi^2(n-1)\beta^{-2}\bar{g}_{\alpha\gamma}, \tag{12}$$

$$\tilde{R}_{\alpha 0} = \tilde{R}_{0\alpha} = \tilde{R}_{00} = 0, \tag{13}$$

$$\tilde{R} = \{\pi^{-2}R - n(n-1)\beta^{-2}\}s^{-2} \tag{14}$$

where α, γ are coordinate indices in $1, \ldots, n$ and R_{ij} and R are the Ricci curvatures and the scalar curvature of $\mathcal{M}$, respectively. See Appendix B for the derivation of such curvatures. The scalar curvature and the Ricci curvatures $\tilde{R}_{\alpha\gamma}$ become more negative than (a constant times of) the original curvature for $\beta < \infty$ and $n \geq 2$. Moreover, the smaller value of β makes the curvature more negative; thus, it becomes possible to control the curvature by tuning β. Note that, the closer to the apex, i.e., the smaller the value of s, the greater the change of the scalar curvature.

Second, assume the original space $\mathcal{M}$ is a length metric space. This does not require a differentiable structure and is more general than the Riemannian manifold. In this case, we cannot argue the curvatures using the Riemannian metric but the CAT(k) property can be used instead. In [24], they proved the curvature of the metric cone is more negative or equal to the curvature of the original space and it can be controlled by β in the sense of the CAT(k) property.

4. Experiments

The claim in this paper is that "a hierarchical structure can be captured by adding a one-dimensional parameter and embedding it in a metric cone." Therefore, we evaluate the proposed method in two experiments:

- Prediction of edge direction for artificially directed graphs;
- Estimation of the hierarchical score by humans for WordNet.

As a comparison, we compare the proposed method with two other methods: Poincaré embedding [17] and ordinary embedding in Euclidean space, which are known to capture the hierarchical structure of graphs. For Euclidean embedding, we use the distance from the mean of embedded data points as the hierarchical score in (8).

4.1. Prediction of Edge Direction for Directed Graphs

In this experiment, we estimate the orientation of directed edges for some simple graphs such that it is natural to think of the direction of the edges as representing the vertical relationship in the hierarchy.

4.1.1. Settings

We use the following three patterns of graphs with a naturally set hierarchical structure:

- Graphs generated by a growing random network model called the Barabási–Albert preferential attachment [31] with $m = 2$, where m is the number of edges to attach from a new node to existing nodes;
- Complete k-ary tree;
- Concatenated tree of two complete k-ary sub-trees.

For the growing random network model, the hierarchy and the corresponding orientation is naturally defined by the order in which each node is attached. For each tree, node depth can be treated as its hierarchy. The concatenated tree is created by connecting the roots of two complete k-ary trees to a new node, which is then used as the new root. The concatenated tree is considered to study the effect of node degree on the cone embedding as will be explained below.

In this experiment, we learn the embedding of each directed graph. However, we use the information of directions only for evaluation and not for learning. For each directed edge of the learned graph, we estimate the direction by the computed hierarchical scores $score(u, v)$:

$$\text{total\_score} = \sum_{(u,v)\in E} \widetilde{score}(u, v)/|E| \tag{15}$$

$$\widetilde{score}(hypo, hype) = \begin{cases} 1 & score(hypo, hype) > 0 \\ 0 & \text{otherwise} \end{cases} \tag{16}$$

where $hype$: higher hierarchy node

$hypo$: lower hierarchy node

Hyperparameters were set as follows. First, the number of negative samplings was set to 5; while increasing the number of negative samplings increases the amount of computation, the effect on accuracy was not significant, so it was set small. Learning was performed for values of β at 0.1, 0.5, 1, and 5, and the best results are described in Table 1. Here all nodes and edges of the graph are used for both training and evaluation.

Table 1. Result for prediction of edge direction for directed graphs. (We list accuracy and standard deviation. Compared to other methods, cone embedding tends to extract the hierarchical structure correctly even when the number of nodes increases).

Model	Barabási–Albert (Nodes: 100)	Complete k-Ary-Tree		Concatenated k-Ary Trees	
		k = 3 (121)	k = 5 (781)	k = 3 (81)	k = 5 (313)
Cone	0.936 (sd: 0.005)	0.787 (0.049)	**0.799** (0.037)	0.783 (0.045)	**0.744** (0.056)
Euclidean	0.181 (0.004)	0.074 (0.088)	0.127 (0.020)	0.190 (0.136)	0.155 (0.031)
Poincaré	**0.957** (0.012)	**0.935** (0.015)	0.351 (0.022)	**0.880** (0.060)	0.606 (0.043)

4.1.2. Results

The experimental results are shown in Table 1 and the cone embedding shows overall good and stable estimation accuracy for hierarchies. Examples in Figure 2 depict how each graph is embedded in a metric cone. Other existing methods may outperform for sparse trees (small degree of each node) but this method has an advantage for dense trees. The reason for the instability in the accuracy of Poincaré embedding may be the lack of invariance with respect to the equidistance transformation, as we have explained. The main reason for the poor hierarchical estimation accuracy of Euclidean embedding is that the root or higher hierarchical nodes are embedded apart from the cluster of other nodes as in Figure 3. As a result, the root node becomes far from the origin of the embedded space.

For the Barabási–Albert model, the relationship between the (added 1D) coordinates of the cone hierarchy and the order is shown in Figure 4. We can see that there is a strong relationship between the degree and the hierarchy. This raises the suspicion that the degree of the node alone determines the hierarchy of the embedding. However, the fact that the cone embedding provides high estimation accuracy even for the concatenated trees with low root degree indicates that this is not true.

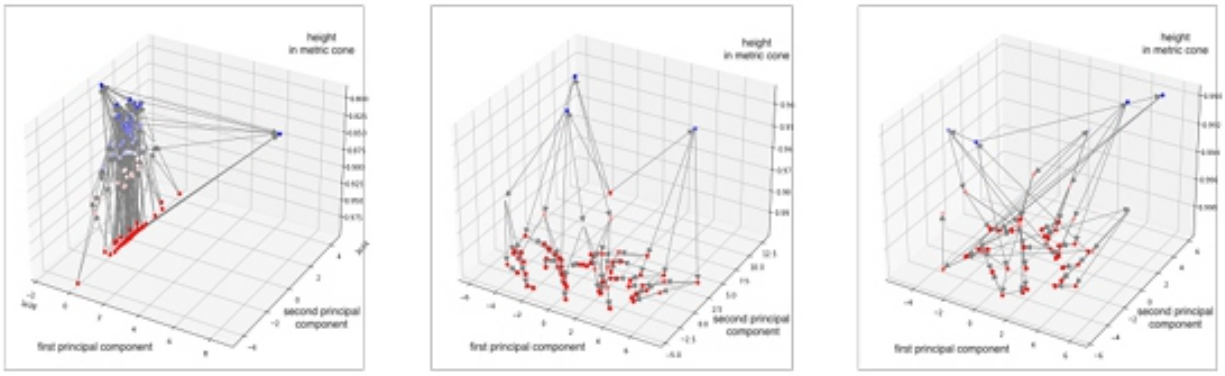


Figure 2. Graphs used for training: (**left**) model trained by Barabási–Albert model, (**middle**) complete k-ary tree, (**right**) concatenated tree of two complete k-ary trees. The x- and y-axes represent embedding in Euclidean space, which is dimensionally reduced to two dimensions by principal component analysis, and the z-axis represents the height in the metric cone (coordinates representing the hierarchy).

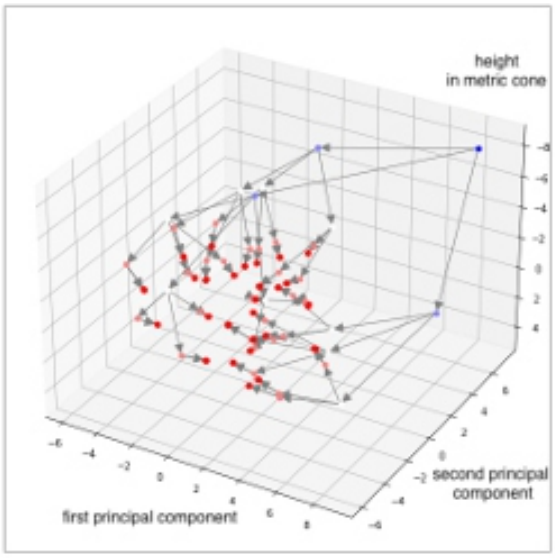


Figure 3. An embedding of the complete k-ary tree ($k = 3$). Each point is plotted by the 3D Euclidean embedding and the color represents length of shortest path from root node (the bluer the color, the higher the hierarchy).

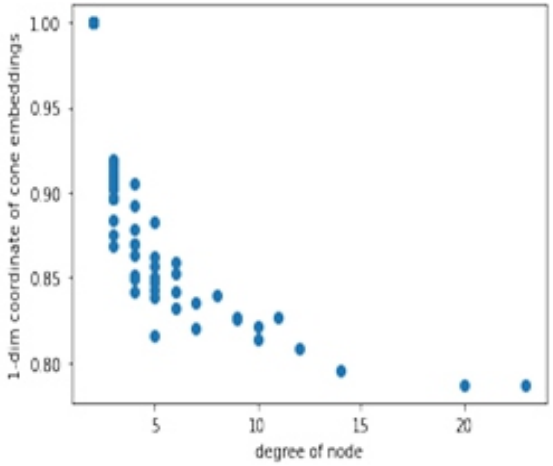


Figure 4. Plot of the hierarchy value of each node in a cone embedding (a newly added one-dimensional parameter) against their node degree.

4.2. Embedding Taxonomies

Following an experiment in [17] for the Poincaré embedding, we evaluate the embedding accuracy of the hierarchical structure using WordNet. To verify this, we embed the nouns in WordNet into a metric cone and use the score function ("total_score" described in Section 4.1). Note that hyperparameter α was set to 10^3. The output of this score function and the correlation coefficient of the HyperLex dataset's score (evaluated manually whether a word is a hyponym of another word) are used to evaluate the ability to represent the hierarchical structure of the model.

In addition to hierarchical scores, we also evaluate the accuracy of graph embedding. We use mean rank and mean average precision, which are commonly used in existing graph embedding accuracy, as evaluation metrics. The mean rank is calculated for each node as the rank of its neighbors when sorted in order of distance. The mean average precision is calculated as follows:

1. Fix one node and calculate the distance to all other nodes.
2. Consider the node adjacent to the fixed node as the correct data, and calculate the average precision for this correct data using the distance as the confidence score.
3. Perform the above two operations on all the nodes and take the average.

The embedding accuracy (mean rank (MR) and mean average precision (MAP)) and correlation coefficients are also shown in Table 2. Note that all of the graph data are used for training and the results are evaluated according to the accuracy with which the graph is reconstructed from the learned embedding. Because the same data are used for training and evaluation, we evaluate the fittingness of the embedding method to the data.

Table 2. MAP, mean rank (MR), Hyperlex score (correlation efficient) and computation time for WordNet. Cone embedding is trained from Euclidean embedding, e.g., in 10-dims cone embeddings, we trained additional 1-dim parameters from 10 dims Euclidean embeddings. For MR and comp. time lower is better, and for MAP and corr. higher is better.

Model	Evaluation Metric	Dimensions			
		10	20	50	100
Euclidean	MR	1681.18	583.75	233.7	162.43
	MAP	0.07	0.12	0.25	0.37
	corr	0.25	0.34	0.38	0.39
	comp. time	976.48	984.72	2169.1	2095.7
Poincaré	MR	1306.22	1183.29	1112.42	1096.08
	MAP	0.09	0.13	0.14	0.16
	corr	0.07	0.08	0.09	0.09
	comp. time	2822.99	1807.73	3954	2241.7
Cone	MR	**426.75**	675.09	777.3	910.51
($\beta = 1.0$)	MAP	**0.10**	0.08	0.07	0.06
	corr	**0.39**	**0.40**	**0.40**	**0.40**
	comp. time	177.94	174.67	**168.33**	**187.83**
Cone	MR	688.85	**143.23**	**74.39**	**51.32**
($\beta = 5.0$)	MAP	0.07	**0.23**	**0.50**	**0.57**
	corr	0.35	0.35	0.37	0.38
	comp. time	**176.04**	**171.21**	168.7	189.89

The table shows that our proposed model improves the score and captures the hierarchical structure better than other embedding methods. Furthermore, "comp. time" represents the time taken to train the embedding (1000 epochs); for cone embedding, it represents the time taken to train in one additional dimension (100 epochs). From the table, we can see that our method is efficient in learning and does not vary with the dimension of the embedding. The result also shows that the optimal β value can depend on the dimension. Because larger β corresponds to smaller curvature, the proposed method seems to perform better when embedding in a smaller curvature space if the dimension becomes higher. For the same reason, Euclidean embedding is considered to perform better than Poincare embedding in high dimensions due to their zero curvature. Tuning of β is necessary because the optimal value also varies depending on the training data. In this case, the search was done in $\beta = 0.1, 0.5, 1.0, 5.0$ but a finer search may improve the accuracy.

Furthermore, an example visualization of the hierarchical structure of the embedding vectors obtained by the training is shown in Figure 5. As the figure illustrates, the closer the coordinate corresponding to the height in the cone is to zero (closer to the top of the cone), the higher the noun in the hierarchy is located in the embedded representation. For visualization, the embedding vectors in Euclidean space were reduced to two dimensions by principal component analysis.

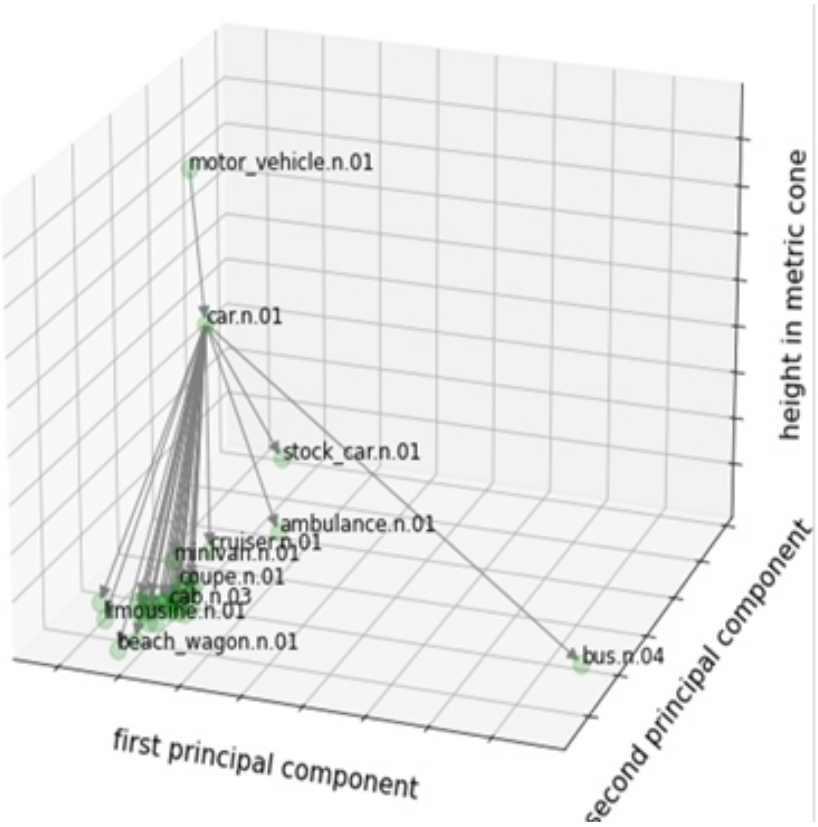
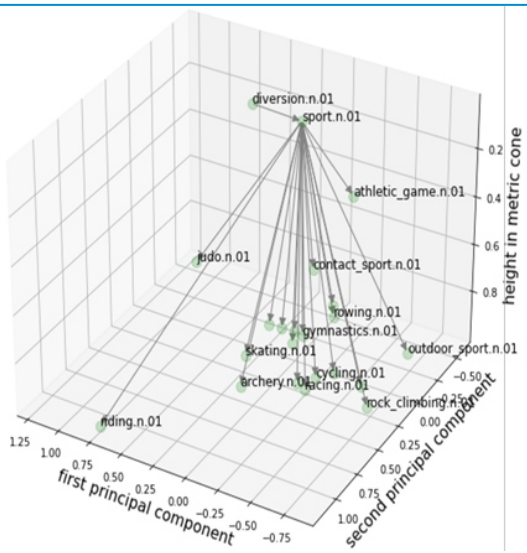


Figure 5. Visualization of wordnet hypernym and hyponym relations by cone embedding (beta: 1.0) learned from 10 to dimensional Euclidean embedding. The right figure is for the car and the left figure is for sport, describing the relationship with one higher or lower word. (In the visualization, all nodes are visualized, but only some of the word names corresponding to each node are shown. The original embedding vector (Euclidean in this case) is also made two-dimensional by PCA.)



5. Discussion and Future Works

In this study, we have demonstrated that a graph embedding in a metric cone that is a dimension larger than the existing embedding methods has the following advantages: (1) we naturally define an index (score function) as an indicator of hierarchy, (2) the proposed method has some adaptivity since it can introduce the hierarchy into various pre-trained models by learning only newly added 1D parameters, and (3) thus, the optimization is computationally inexpensive and stable. By optimizing the 1D parameters, we have shown that the proposed method also has the flexibility to optimize the curvature to enhance the accuracy as well as other methods. Since the metric cone is defined as a space with +1 dimension with respect to the original metric space, it is also possible to learn cone embedding in the same way even when the original space is Poincaré. We demonstrated the feasibility of extracting the hierarchical structure using solely the additional space by fixing the original space and learning its embedding.

It is worth noting that an alternative approach involves directly embedding the graph into the metric cone by learning an embedding that includes the source space. However, the constraint of learning in one dimension offers some advantages. For instance, the metric cone is defined as a space with +1 dimension with respect to the original metric space, which allows for the learning of cone embedding in the same way for any general original spaces, including the Poincaré space. The independence of the optimization algorithm from the original embedding space results in a more stable computation. For example, the cone embedding performed best or fairly for overall settings while the Poincaré embedding performed very poorly in some experimental settings. Moreover, the tuning of hyperparameters such as β for embedding can be performed independently from the original embedding.

Author Contributions: Conceptualization, K.K.; Methodology, K.K.; Formal analysis, D.T.; Investigation, D.T.; Writing—original draft, D.T.; Writing—review & editing, K.K.; Supervision, K.K. All authors have read and agreed to the published version of the manuscript.

Funding: This work was supported by RIKEN AIP and JSPS KAKENHI (JP22K03439, JP19K00912).

Data Availability Statement: Tree-like graph data were randomly generated by the python library networkx (https://networkx.org/). WordNet (https://www.nltk.org/) was obtained from the python library NLTK (https://www.nltk.org/) and used for training. For the human numerical evaluation results used in the evaluation of the hierarchical scores, hyperlex (https://github.com/cambridgeltl/hyperlex) was used.

Acknowledgments: The idea of applying metric cones to data science was born during a collaboration with Henry P. Wynn.

Conflicts of Interest: The authors declare no conflict of interest.

Appendix A. Derivation of the Metric Tensor of a Metric Cone

Let $\mathcal{M}$ be an n-dimensional Riemannian manifold with a metric g. Then the metric $\bar{g}$ of the corresponding metric cone $\bar{\mathcal{M}} = \bar{\mathcal{M}}_\beta$ can be defined except the apex. Denote the square of the infinitesimal distance in $\bar{\mathcal{M}}$ as $|d\bar{s}|^2$, then

$$\begin{aligned}
|d\bar{s}|^2 &= \bar{d}_\beta((x,r),(x+dx,r+dr))^2 \\
&= \beta^2\left(2r^2 + 2rdr + dr^2 - 2(r+dr)r\cos(\pi \min(d_{\mathcal{M}}(x,x+dx)/\beta, 1))\right) \\
&\approx \beta^2\left(2r^2 + 2rdr + dr^2 - 2(r^2+rdr)\left(1 - \frac{(\pi d_{\mathcal{M}}(x,x+dx)/\beta)^2}{2}\right)\right) \\
&\approx \beta^2 dr^2 + \pi^2 r^2 \sum_{i,j} g_{ij}dx_i dx_j + \pi^2 r dr \sum_{i,j} g_{ij}dx_i dx_j \\
&= \begin{pmatrix} dx \\ dr \end{pmatrix}^\top \begin{pmatrix} (\pi^2 r^2 g_{ij}) & 0 \\ 0 & \beta^2 \end{pmatrix} \begin{pmatrix} dx \\ dr \end{pmatrix}. \qquad \text{(A1)}
\end{aligned}$$

Therefore, the metric tensor $\bar{g}$ becomes

$$\bar{g} = \begin{pmatrix} \pi^2 r^2 g & 0 \\ 0 & \beta^2 \end{pmatrix}. \qquad \text{(A2)}$$

Appendix B. Derivation of the Ricci and the Scalar Curvatures of a Metric Cone

We will derive the Ricci and scalar curvatures of metric cone $\bar{\mathcal{M}}$ Let $0, 1, \ldots, n$ be the coordinate indices of metric cone $\bar{\mathcal{M}}$ where 0 corresponds to the radial coordinate $s \in (0,1)$ and $1, \ldots, n$ correspond to $x \in \mathcal{M}$.

Claim A1. *The Ricci curvatures $\bar{R}_{ij}$ and the scalar curvature $\bar{R}$ become*

$$\bar{R}_{\alpha\gamma} = R_{\alpha\gamma} - \pi^2(n-1)\beta^{-2}g_{\alpha\gamma}, \bar{R}_{\alpha 0} = \bar{R}_{0\alpha} = \bar{R}_{00} = 0,$$

$$\bar{R} = \{\pi^{-2}R - n(n-1)\beta^{-2}\}s^{-2} \qquad \text{(A3)}$$

where α and γ are coordinate indices in $1, \ldots, n$ and R_{ij} and R are the Ricci curvatures and the scalar curvature of $\mathcal{M}$, respectively.

Proof. By Example 4.6 of [32], if the metric of $\bar{\mathcal{M}}$ is defined by the squared infinitesimal distance $|ds|^2$ in $\mathcal{M}$ and a $\mathcal{C}^2$-class function w on an open interval $J \subset \mathbb{R}$ as

$$|d\bar{s}|^2 = \beta^2|dr|^2 + w(r)^2|ds|^2, \qquad \text{(A4)}$$

the Ricci curvature tensor becomes

$$\tilde{R}_{\alpha\gamma} = R_{\alpha\gamma} - \left((n-1)\left(\frac{w'}{w}\right)^2 + \frac{w''}{w}\right)\tilde{g}_{\alpha\gamma} = R_{\alpha\gamma} - \left((n-1)\left(\frac{w'}{w}\right)^2 + \frac{w''}{w}\right)w^2 g_{\alpha\gamma},$$

$$\tilde{R}_{\alpha 0} = 0, \quad \tilde{R}_{00} = -(n-1)\frac{w''}{w} \tag{A5}$$

and the scalar curvature becomes

$$\tilde{R} = w^{-2}(R - n(n-1)(w')^2 - 2nww''). \tag{A6}$$

Since the metric of a metric cone $\tilde{M}$ is given by

$$|d\bar{s}|^2 = \beta^2|dr|^2 + \pi^2 r^2|ds|^2, \tag{A7}$$

by setting $\bar{r} := \beta r$ and $w(\bar{r}) := \pi\beta^{-1}\bar{r}$, we obtain the following form similar to (A4):

$$|d\bar{s}|^2 = |d\bar{r}|^2 + w(\bar{r})^2|ds|^2. \tag{A8}$$

By substituting $w(\bar{r}) = \pi\beta^{-1}\bar{r} = \pi r$, $w'(\bar{r}) = \pi\beta^{-1}$ and $w''(\bar{r}) = 0$, we obtain the Ricci and scalar curvatures in Claim A1. □

Appendix C. Identifiability of the Heights in the Cone Embedding

In this section, we will prove Theorem 1 of the main article. Let us begin by rewriting Theorem 1 as a longer but more theoretically rigorous form.

Theorem A1 (A rigorous restatement of Theorem 1)**.** *Let Z be a length metric space and X be a metric cone of Z with a parameter $\beta > 0$. Let n be an integer at least 3. Fix $z_i \in Z$ and $x_i := (z_i, t_i) \in X$ with $t_i \in [0,1]$ for $i = 1, \ldots, n$. Denote a matrix $\tilde{D} := [\tilde{d}_\beta(x_i, x_j)]_{i,j=1}^n$.*

(a) Assume $z_1, \ldots, z_n$ are not all aligned in a geodesic. Given $z_1, \ldots, z_n$ and $\tilde{D}$, the number of possible values of $(t_1, \ldots, t_n)$ is at most four.

(b) Let $n \geq 4$ and assume $z_1, \ldots, z_n$ and $t_1, \ldots, t_n$ are in a "general" position. Here "general" position means that, besides the assumption in (a), given any four distinct points $z_i, z_j, z_k, z_l \in Z$ and corresponding heights $t_i, t_j, t_k \in [0,1]$, t_l can still take infinitely many values. Then $t_1, \ldots, t_n$ are determined uniquely by $z_1, \ldots, z_n$ and $\tilde{D}$.

(c) If $d(z_i, z_j) \geq \beta/2$ for all $i, j = 1, \ldots, n$, $i \neq j$, then $t_1, \ldots, t_n$ are determined uniquely by $z_1, \ldots, z_n$ and $\tilde{D}$.

Before the proof, we will state some remarks.

If $n = 2$, the identifiability problem reduces to an elementary geometric question: given a circle sector as the right two subfigures of Figure 1 of the main paper and the length of the blue line segment(s) connecting (x, s) and (y, t), can s and t be determined uniquely? The answer is evidently no. However, it is notable that there are two types of counterexamples. The first type is as Figure A1a, one point moves "up" and the other moves "down". The other type as Figure A1b is maybe counter-intuitive: both move "up" or "down". Note that the second case does not happen if the angle θ is larger than or equal to $\pi/2$.

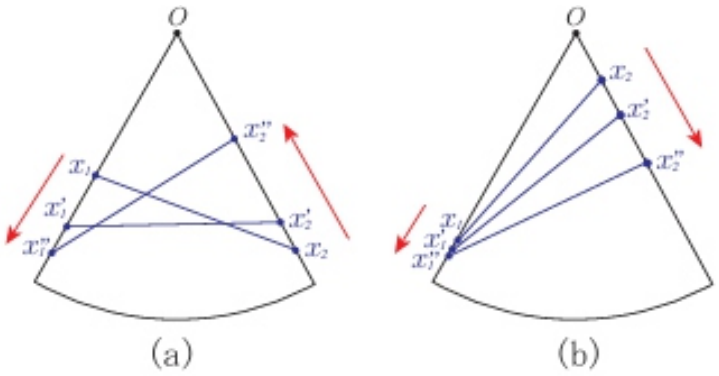


Figure A1. Two types of movement for a line segment of constant length (**a**) One point moves "up" and the other moves "down" (**b**) Both move "up" or "down".

If $n = 3$, the picture becomes a tetrahedron as in Figure A2. Here the angles and edge lengths are defined by

$$\begin{aligned} \theta_1 &:= \pi \min(d_Z(z_2, z_3)/\beta, 1), \quad a_1 := \tilde{d}_\beta(x_2, x_3) \\ \theta_2 &:= \pi \min(d_Z(z_3, z_1)/\beta, 1), \quad a_2 := \tilde{d}_\beta(x_3, x_1) \\ \theta_3 &:= \pi \min(d_Z(z_1, z_2)/\beta, 1), \quad a_3 := \tilde{d}_\beta(x_1, x_2) \end{aligned} \tag{A9}$$

and $\theta_1 + \theta_2 + \theta_3$ is assumed to be at most 2π. Then, the geometrical question becomes "when angles α, β, γ and edge lengths a_1, a_2, a_3 of triangle $\triangle x_1 x_2 x_3$ is given, can the position of the points x_1, x_2, and x_3 be determined uniquely?" If it is not unique and there are two different positions of x_1, x_2, and x_3, at least one edge should move as in Figure A1b since it is impossible to move all three edges as in Figure A1a. However, if all of the angles are larger than or equal to $\pi/2$, this cannot happen. This actually gives a geometrical proof of Theorem A1(c).

If $\theta_1 + \theta_2 + \theta_3$ is larger than 2π, the geometric arguments become complicated. We do not need this kind of case analysis when we use algebraic arguments as in the following proof.

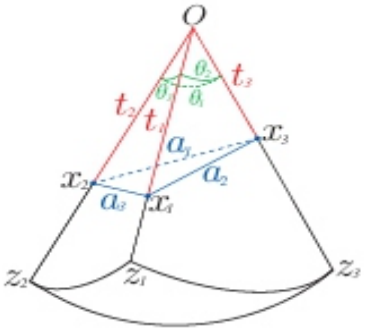


Figure A2. Metric cone generated by three points $z_1, z_2, z_3 \in Z$.

Now we will prove the theorem. In the proof, we use the Gröbner basis as a tool of computational algebra. See for example [33] about definition and application of the Gröbner basis.

Proof. (a) Since the maximum number of possible values of $(t_1, \ldots, t_n)$ does not increase with n, it is enough to prove for $n = 3$. We set $\theta_1, \theta_2, \theta_3 \in [0, \pi]$ and $a_1, a_2, a_3 \geq 0$ as (A9). Then, by the law of cosine,

$$
\begin{aligned}
t_2^2 + t_3^2 - 2t_2t_3\cos\theta_1 &= a_1^2,\\
t_3^2 + t_1^2 - 2t_3t_1\cos\theta_2 &= a_2^2,\\
t_1^2 + t_2^2 - 2t_1t_2\cos\theta_3 &= a_3^2.
\end{aligned} \tag{A10}
$$

We consider this as a system of polynomial equations with variables t_1, t_2, t_3 and compute the Gröbner basis of the ideal generated by the corresponding polynomials by degree lexicographic monomial order (deglex) with $t_1 > t_2 > t_3$ by Mathematica (see Note A1). Then the output becomes as in Note A1 and the basis includes $-t_1^2 + (2\cos\theta_2)t_1t_3 - t_3^2 + a_2^2$, $-t_2^2 + (2\cos\theta_3)t_2t_3 - t_3^2 + a_3^2$ and $4v(\theta_1,\theta_2,\theta_3)t_3^4 + (\text{terms of degree} \leq 2)$ where

$$
v(\theta_1,\theta_2,\theta_3) := 1 + 2\cos\theta_1\cos\theta_2\cos\theta_3 - \cos\theta_1^2 - \cos\theta_2^2 - \cos\theta_3^2. \tag{A11}
$$

Note that when $\theta_1 + \theta_2 + \theta_3 \leq 2\pi$, $\frac{a_1a_2a_3}{6}v(\theta_1,\theta_2,\theta_3)$ is a formula of the volume of the tetrahedron whose base triangle is $\triangle x_1x_2x_3$ and, therefore, it has a positive value unless the tetrahedron degenerates. By the assumption, z_1, z_2, z_3 are not aligned in a geodesic and therefore the tetrahedron does not degenerate and $v(\theta_1,\theta_2,\theta_3)$ must be nonzero. Note that this becomes negative when $\theta_1 + \theta_2 + \theta_3 > 2\pi$.

On the other hand, it is known that the system of polynomial equations with a Gröbner basis G has a finite number of (complex) solutions if and only if, for each variable x, G contains a polynomial with a leading monomial that is a power of x. Now, all variables t_1,t_2, and t_3 satisfy such a property; thus, we conclude there are at most a finite number of solutions.

Then, by Bézout's theorem, the number of solutions is at most the product of the degree of three polynomial equations, i.e., $2 \times 2 \times 2 = 8$. However, if (t_1, t_2, t_3) is a solution, $(-t_1, -t_2, -t_3)$ is also a solution, and only one of each pair can satisfy $t_1, t_2, t_3 \leq 0$. Thus, we conclude the number of possible values of (t_1, t_2, t_3) is at most four.

(b) By the assumptions in (a), without loss of generality, we can assume z_1, z_2, z_3 are not aligned in a geodesic. By the result of (a), given z_1, z_2, z_3 and distances $\bar{d}_\beta(x_1, x_2)$, $\bar{d}_\beta(x_1, x_3)$,$\bar{d}_\beta(x_2, x_3)$, there are at most four variations of the values of (t_1, t_2, t_3). Here we assume t_1 can take multiple values including $\hat{t}_1$ and $\check{t}_1$.

Suppose, in addition to the above, the values of z_4 and $d_\beta(x_1, x_4)(=: a_4)$ are given and let $\theta_4 := \pi\min(d_Z(z_1, z_4)/\beta, 1)$. Then, both $\hat{t}_1$ and $\check{t}_1$ satisfy $t_1^2 + t_4^2 - 2t_1t_4\cos\theta_4 = a_4^2$ and therefore $2t_4\cos\theta_4 = \hat{t}_1 + \check{t}_1$ must hold. Since $\hat{t}_1$ and $\check{t}_1$ are different non-negative values, $\hat{t}_1 + \check{t}_1 > 0$ and, therefore, $\cos\theta_4 \neq 0$. Hence, we obtain $t_4 = (\hat{t}_1 + \check{t}_1)/2\cos\theta$.

This means if t_4 takes values except $(\hat{t}_1 + \check{t}_1)/2\cos\theta$, at most only one of $\hat{t}_1$ and $\check{t}_1$ can be a solution. We can reduce each pairwise ambiguity of the (at most) four possibilities of (t_1, t_2, t_3) one by one similarly. Finally (t_1, t_2, t_3) are determined uniquely for all except at most $\binom{4}{2} = 6$ values of t_4. However, such finite values of t_4 can be neglected thanks to the assumption of the "general" position in the theorem. Since the same argument holds for any triplets, the statement has been proved.

(c) If $(t_1, \ldots, t_n)$ can take multiple values, without loss of generality we can assume (t_1, t_2, t_3) takes multiple values. By the assumption in the theorem, $\theta_1, \theta_2, \theta_3 \geq \pi/2$ and therefore all coefficients in each equation of (A10) become positive. Thus, if t_i increases/decreases then t_j must decrease/increase for $(i, j) = (1,2), (2,3), (3,1)$ but this cannot happen simultaneously. Hence, (t_1, t_2, t_3) cannot take multiple values.

Note that all of this proof works even when $\theta_1 + \theta_2 + \theta_3$ is larger than 2π. □

Remark A1. *The assumption in Theorem A1(a) is necessary. If the assumption fails, the tetrahedron degenerates and x_1, x_2, x_3 and the apex O are all in a plane. When O happens to be on a circle passing through x_1, x_2, and x_3, move O to another point O' on the same circle. Then, the angles corresponding to $\theta_1, \theta_2, \theta_3$ do not change by the inscribed angle theorem. By an elemental geometrical argument, a new position of x_1, x_2, x_3 and O' gives another solution of t_1, t_2, t_3. Hence, obviously, there are an infinite number of solutions.*

Remark A2. *The assumption of "general" positions of $z_1, \ldots, z_n$ in Theorem A1(b) is satisfied easily for most data distributions. For example, if both $z_1, \ldots, z_n \in \mathbb{R}^d$ and $t_1, \ldots, t_n \in [0, 1]$ are i.i.d. from a probability distribution whose density function exists with respect to the Lebesgue measure, then it is easy to see the assumption holds almost surely and therefore uniqueness of the solution is guaranteed. Note that, for $n = 3$ under the same setting, there can be multiple solutions with a positive probability.*

Note A1. *Computation of the Gröbner basis by Mathematica:*

For simplicity, we put $x := t_1$, $y := t_2$, $z := t_3$, $a := 2\cos\theta_1$, $b := 2\cos\theta_2$, $c := 2\cos\theta_3$, $d := a_1^2$, $e := a_2^2$, and $f := a_3^2$.

Note that the second, first, and last polynomials in the output correspond to $-t_1^2 + (2\cos\theta_2)t_1t_3 - t_3^2 + a_2^2$, $-t_2^2 + (2\cos\theta_3)t_2t_3 - t_3^2 + a_3^2$, and $4v(\theta_1, \theta_2, \theta_3)t_3^4 + (\text{terms of degree} \leq 2)$ in the proof, respectively.

```
------------------------------------------------------------
In := GroebnerBasis[{x^2 + y^2 - a*x*y - d, x^2 + z^2 - b*x*z - e,
y^2 + z^2 - c*y*z - f}, {x, y, z},
MonomialOrder -> DegreeLexicographic]

Out = {f - y^2 + c y z - z^2, e - x^2 + b x z - z^2,
d - x^2 + a x y - y^2,
d x - e x + a e y - x y^2 - b d z + b y^2 z + x z^2 -
a y z^2, -c d x + c e x - a c e y + b f y + c x y^2 - b y^3 +
b c d z - a f z + a y^2 z - c x z^2 - b y z^2 + a z^3,
a f x + d y - f y - x^2 y - c d z + c x^2 z - a x z^2 + y z^2,
b f x - c e y + c x^2 y - b x y^2 + e z - f z - x^2 z +
y^2 z, -c e x + a b f x + c x^3 + b d y - b f y - b x^2 y -
b c d z + a e z - a x^2 z + c x z^2 + b y z^2 - a z^3,
a c d x - a c e x + b f x + c d y - c e y + a^2 c e y - a b f y -
b x y^2 + a b y^3 - c y^3 - a b c d z + e z - f z + a^2 f z -
x^2 z + y^2 z - a^2 y^2 z + a c x z^2 + a b y z^2 -
a^2 z^3, -a e f - d x y + e x y + f x y + c d x z - c e x z +
b d y z - b f y z - b c d z^2 + a e z^2 + a f z^2 - 2 x y z^2 +
c x z^3 + b y z^3 - a z^4, -c d e + c e^2 - a b e f + c d x^2 -
c e x^2 - b d x y + b e x y + b f x y - a f x z - d y z +
b^2 d y z + 2 e y z + f y z - b^2 f y z - x^2 y z + 2 c d z^2 -
b^2 c d z^2 + a b e z^2 - 2 c e z^2 + a b f z^2 + a x z^3 -
3 y z^3 + b^2 y z^3 - a b z^4 + c z^4, -d x + a^2 d x + a b c d x +
2 e x - a^2 e x - a b c e x - a^2 f x + b^2 f x - x^3 + b c d y -
a e y + a^3 e y - b c e y + a^2 b c e y + a f y - a b^2 f y +
x y^2 - b^2 x y^2 - a y^3 + a b^2 y^3 - b c y^3 + b d z -
```

```
a^2 b d z + a c d z - a b^2 c d z + b e z - a c e z - b f z +
a^2 b f z - 2 x z^2 + 2 a^2 x z^2 - a^3 y z^2 + a b^2 y z^2 -
a^2 b z^3 + a c z^3, -c d^2 + c d e + a b d f + c d x^2 - c e x^2 +
b f x y - a b d y^2 + 2 c d y^2 - 2 c e y^2 + a^2 c e y^2 -
a b f y^2 - b x y^3 + a b y^4 - c y^4 - a d x z + a e x z -
a f x z - 2 d y z + e y z - a^2 e y z - f y z + a^2 f y z +
x^2 y z + 3 y^3 z - a^2 y^3 z,
d^2 - 2 d e + c^2 d e + e^2 - c^2 e^2 - 2 d f + b^2 d f + 2 e f -
a^2 e f + a b c e f + f^2 - b^2 f^2 - c^2 d x^2 + c^2 e x^2 +
b c d x y - b c e x y - b c f x y - b^2 d y^2 + b^2 f y^2 +
a c d x z - a c e x z + a c f x z + a b d y z + a b e y z -
a b f y z + 4 d z^2 - 2 b^2 d z^2 - a b c d z^2 - 2 c^2 d z^2 +
b^2 c^2 d z^2 - 4 e z^2 + a^2 e z^2 - a b c e z^2 + 2 c^2 e z^2 -
4 f z^2 + a^2 f z^2 + 2 b^2 f z^2 - a b c f z^2 + 4 z^4 - a^2 z^4 -
b^2 z^4 + a b c z^4 - c^2 z^4}
```

References

1. Zhang, J.; Ackerman, M.S.; Adamic, L. Expertise networks in online communities: Structure and algorithms. In Proceedings of the 16th international Conference on World Wide Web, Banff, AB, Canada, 8–12 May 2007; pp. 221–230.

2. DeChoudhury, M.; Counts, S.; Horvitz, E. Social media as a measurement tool of depression in populations. In Proceedings of the 5th Annual ACM WebScience Conference, Paris, France, 2–4 May 2013; pp. 47–56.

3. Page, L.; Brin, S.; Motwani, R.; Winograd, T. The PageRank Citation Ranking: Bringing Order to the Web; Technical Report; Stanford InfoLab, Stanford University: Stanford, CA, USA, 1999.

4. Barabasi, A.L.; Oltvai, Z.N. Network biology: Understanding the cell's functional organization. Nat. Rev. Genet. 2004, 5, 101–113. [CrossRef] [PubMed]

5. Yahya, M.; Berberich, K.; Elbassuoni, S.; Weikum, G. Robust question answering over the web of linked data. In Proceedings of the 22nd ACMInternational Conference on Information & Knowledge Management, San Francisco, CA, USA, 27 October–1 November 2013; pp. 1107–1116.

6. Hoffart, J.; Milchevski, D.; Weikum, G. STICS: Searching with strings, things, and cats. In Proceedings of the 37th International ACMSIGIRConference on Research & Development in Information Retrieval, Gold Coast, Queensland, Australia, 6–11 July 2014; pp. 1247–1248.

7. Klimovskaia, A.; Lopez-Paz, D.; Bottou, L.; Nickel, M. Poincaré maps for analyzing complex hierarchies in single-cell data. Nat. Commun. 2020, 11, 2966. [CrossRef] [PubMed]

8. Kipf, T.N.; Welling, M. Semi-supervised classification with graph convolutional networks. arXiv 2016, arXiv:1609.02907.

9. Ribeiro, L.F.; Saverese, P.H.; Figueiredo, D.R. struc2vec: Learning node representations from structural identity. In Proceedings of the 23rd ACM SIGKDDInternational Conference on Knowledge Discovery and Data Mining, Halifax, NS, Canada, 13–17 August 2017; pp. 385–394.

10. Hamilton, W.; Ying, Z.; Leskovec, J. Inductive representation learning on large graphs. Adv. Neural Inf. Process. Syst. 2017, 30, 1025–1035.

11. Goyal, P.; Ferrara, E. Graph embedding techniques, applications, and performance: A survey. Knowl. Based Syst. 2018, 151, 78–94. [CrossRef]

12. Grover, A.; Leskovec, J. node2vec: Scalable feature learning for networks. In Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, San Francisco, CA, USA, 13–17 August 2016; pp. 855–864.

13. Cao,S.; Lu, W.; Xu, Q. Grarep: Learning graph representations with global structural information. In Proceedings of the 24th ACMInternational on Conference on Information and Knowledge Management, Melbourne, VIC, Australia, 19–23 October 2015; pp. 891–900.

14. Tang, J.; Qu, M.; Wang, M.; Zhang, M.; Yan, J.; Mei, Q. Line: Large-scale information network embedding. In Proceedings of the 24th International Conference on World Wide Web, Florence, Italy, 18–22 May 2015; pp. 1067–1077.

15. Sun, Z.; Chen, M.; Hu, W.; Wang, C.; Dai, J.; Zhang, W. Knowledge Association with Hyperbolic Knowledge Graph Embeddings. In Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP), Association for Computational Linguistics, Online, 16–20 November 2020; pp. 5704–5716. [CrossRef]

16. Rezaabad, A.L.; Kalantari, R.; Vishwanath, S.; Zhou, M.; Tamir, J. Hyperbolic graph embedding with enhanced semi-implicit variational inference. In Proceedings of the International Conference on Artificial Intelligence and Statistics, PMLR, Virtual, 13–15 April 2021; pp. 3439–3447.

17. Nickel, M.; Kiela, D. Poincaré embeddings for learning hierarchical representations. In Proceedings of the Advances in Neural Information Processing Systems, Long Beach, CA, USA, 4–9 December 2017; pp. 6338–6347.

18. Zhang,Z.; Cai, J.; Zhang, Y.; Wang, J. Learning hierarchy-aware knowledge graph embeddings for link prediction. In Proceedings of the AAAI Conference on Artificial Intelligence, New York, NY, USA, 7–12 February 2020; Volume 34, pp. 3065–3072.

19. Chami, I.; Wolf, A.; Juan, D.C.; Sala, F.; Ravi, S.; Ré, C. Low-Dimensional Hyperbolic Knowledge Graph Embeddings. In Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics, Association for Computational Linguistics, Online, 5–10 July 2020; pp. 6901–6914. [CrossRef]

20. Dhingra, B.; Shallue, C.; Norouzi, M.; Dai, A.; Dahl, G. Embedding Text in Hyperbolic Spaces. In Proceedings of the Twelfth Workshop on Graph-Based Methods for Natural Language Processing (TextGraphs-12),Association for Computational Linguistics, NewOrleans, LA, USA, 6 June 2018; pp. 59–69. [CrossRef]

21. Nickel, M.; Kiela, D. Learning Continuous Hierarchies in the Lorentz Model of Hyperbolic

Geometry. In Proceedings of the Machine Learning Research, PMLR, Stockholmsmässan, Stockholm, Sweden, 10–15 July 2018; Volume 80, pp. 3779–3788.

22. Ganea, O.; Becigneul, G.; Hofmann, T. Hyperbolic Entailment Cones for Learning Hierarchical Embeddings. In Proceedings of the Machine Learning Research, PMLR, Stockholmsmässan, Stockholm, Sweden, 10–15 July 2018; Volume 80, pp. 1646–1655.

23. Sala, F.; De Sa, C.; Gu, A.; Ré, C. Representation tradeoffs for hyperbolic embeddings. In Proceedings of the International Conference on Machine Learning, PMLR, Stockholm, Sweden, 10–15 July 2018; pp. 4460–4469.

24. Kobayashi, K.; Wynn, H.P. Empirical geodesic graphs and CAT (k) metrics for data analysis. Stat. Comput. 2020, 30, 1–18. [CrossRef]

25. Wilson, R.C.; Hancock, E.R.; Pekalska, E.; Duin, R.P. Spherical and Hyperbolic Embeddings of Data. IEEE Trans. Pattern Anal. Mach. Intell. 2014, 36, 2255–2269. [CrossRef] [PubMed]

26. Chami,I.; Ying, Z.; Ré, C.; Leskovec, J. Hyperbolic Graph Convolutional Neural Networks. In Proceedings of the Advances in Neural Information Processing Systems; Wallach, H., Larochelle, H., Beygelzimer, A., d'Alché-Buc, F., Fox, E., Garnett, R., Eds.; Curran Associates, Inc.: Red Hook, NY, USA, 2019; Volume 32.

27. Sturm, K.T. Probability measures on metric spaces of nonpositive curvature. In Proceedings of the Heat Kernels and Analysis on Manifolds, Graphs, and Metric Spaces: Lecture Notes A Quart, Program Heat Kernels, Random Walks, Analysis Manifolds Graphs, Emile Borel Cent. Henri Poincaré Institute, Paris, France, 16 April–13 July 2002 ; Volume 338, p. 357. Available online: https://bookstore.ams.org/conm-338 (accessed on 24 February 2023).

28. Deza, M.M.; Deza, E. Encyclopedia of Distances; Springer: Berlin/Heidelberg, Germany, 2009; pp. 1–583. 29. Loustau, B. Hyperbolic geometry. arXiv 2020, arXiv:2003.11180.

30. Sarkar, R. Low distortion delaunay embedding of trees in hyperbolic plane. In Proceedings of the International Symposium on Graph Drawing, Eindhoven, The Netherlands, 21–23 September 2011; pp. 355–366.

31. Barabasi, A.L.; Albert, R. Emergence of Scaling in Random Networks. Science 1999, 286, 509–512. [CrossRef] [PubMed]

32. Janson, S. Riemannian geometry: Some examples, including map projections. Notes. 2015. Available online: http://www2.math. uu.se/~svante/papers/sjN15.pdf (accessed on 24 February 2023).

.

33. Cox,D.; Little, J.; OShea, D. Ideals, Varieties, and Algorithms: An Introduction to Computational Algebraic Geometry and Commutative Algebra; Springer Science & Business Media: Berlin/Heidelberg, Germany, 2013.

Large-Signal Stability of the Quadratic Boost Converter Using a Disturbance Observer-Based Sliding-Mode Control

Satyajit Chincholkar 1,*, Mohd Tariq and Shabana Urooj 3

Department of Electronics and Telecommunication Engineering, School of E&TC Engineering, MIT Academy of Engineering, Pune 412105, India Department of Electrical Engineering, ZHCET, Aligarh Muslim University, Aligarh 202002, India; Department of Electrical Engineering, College of Engineering, Princess Nourah bint Abdulrahman University P.O. Box 84428, Riyadh 11671, Saudi Arabia Correspondence:

ABSTRACT

The quadratic boost (QB) converter is a fourth-order system with a dc gain that is higher than the traditional second-order step-up configuration. The modern controllers that control these high-order dc–dc converters often only guarantee local stability around a steady-state equilibrium point, which is one of their primary drawbacks. In this article, a non-linear robust control law design to attain large-signal stability in this single switch QB converter is presented. In the presence of an unpredictable load, the control objective is to maintain the regulation of an output voltage. The Brunovsky canonical model of the converter was derived first, and the non-linear disturbance observer-based sliding-mode (SM) control law is designed based on it. An observer variable precisely estimates the output disturbances. The detailed process for deriving the control signal is described in this paper and the large-signal stability of the closed-loop converter system is ensured via the Lyapunov function. Finally, some simulation results are shown to validate the usefulness of the given controller.

Keywords: quadratic converter; sliding-mode control; observer

Introduction

The dc–dc boost converter is employed in several fields, including electric vehicles, telecommunication equipment, energy systems based on non-conventional resources, and so on [1–3]. For instance, the voltage at the output of a single fuel cell is very low, of the order of 1.1 V, and its stacked version could produce around from 24 V to 60 V. However, this voltage is not enough at the input of an inverter for applications in the power range from 1 kWto5kW.Thus,adc–dcconverter can be employed between the nonconventional energy resource and inverter and its gain should be high enough to make up for the differences [1]. The second-order classical boost converter can step up the output voltage, but its gain is limited because it needs to operate at a considerably high duty ratio to produce high gain, and switching devices have limited finite switching durations. It may also incur EMI and reverse recovery issues of the diode. Lastly, working at high duty ratio values could affect the system's dynamic response to parameter variations [4]. One of the solutions to address this problem is using transformer-based dc–dc power converters to provide high gain before interfacing with inverters. However, if a particular utilization area does not need any isolation, the usage transformer becomes redundant, and it ultimately increases the system's size and pricing. The high-order transformer-less power converters are thus receiving attention because they do not only eliminate the use of transformers but also avoid high values of the

duty ratio to offer a large conversion ratio [5–8]. Moreover the voltage stress is also limited in their cases. Among them, the quadratic boost (QB) converter is a popular candidate due to its high efficiency, smaller size, and ease of control due to having a single active switch [6]. The control aspect of high-step-up converters like the QB converter has recently been the interest of some research [6–12]. However, their control is not very straight forward because right half-plane zeroes are present in the control to load voltage transfer function of these converters [13]. Because of this, the closed-loop system could lose its stability. To address the first concern, one of the widely used regulation methods for the higher order power converters is employing the current through the inductor for feedback purposes [14]. The additional current-loop, apart from the basic voltage-loop, provides stability to the system andgivesinbuilt overcurrent protection. In [13], the application of the current-based control scheme for the quadratic boost converter was investigated. The more advanced current-mode control, based on an adaptation algorithm for the sixth-order boost converter, was discussed in [15]. Even though all of these current-mode controllers are shown to provide a satisfactory response over a large range of parameter changes, they are based on the small-signal averaged model of the converter, which can only ensure stability in the vicinity of a steady-state operating point. Asliding-mode (SM) scheme is another well-employed scheme that is suitable for dcdc converters [16–24]. Traditionally, hysteresis-modulation is used for the implementation of the SM controller for dc–dc converters [6,20–22]. This method has recently been used to regulate the output voltage of several high-gain converters like the quadratic boost converter [6,20], the hybrid boost converter [21], and the zeta converter [22]. Although this method is easy to implement, its main drawback is that it may lead to chattering in the response. Also, since the switching frequency is not fixed, there could be large variations in the switching frequency in the presence of load and line variations. This maylead to increased switching losses and electromagnetic interference (EMI) issues. To address these concerns, recently, the constant-frequency SM scheme has been employed for high-order boost converters like the quadratic boost converter in [16]. In this method, the pulse-width-modulation (PWM) technique is used to generate the control signal. The various advantages offered by this method are ease of implementation, reduced chattering, and lower electromagnetic interference (EMI) issues. Some of the other state-of-the-art SM controllers for dc–dc converters based on the PWM approach are discussed in [17,18]. As can beseen, there has been considerable efforts made to wards the implementation of several non-linear controllers for high-order dc–dc converters. However, the main drawback of most of these controllers is that their stability is guaranteed only in the neighborhood of the equilibrium point. In other words, they guarantee only small-signal stability and none of the works discussed so far address the large-signal stability of the controlled high-stepup power converters. Thus, to ensure smooth tracking in the presence of large and fast variations in the system parameters, the problem of the design of a robust and globally stable controller for high-step-up dc–dc converters still needs to be addressed. To address this, a new SM controller design based on disturbance observer (DO) for the QB converter's output voltage regulation is

presented. The main contributions of the paper are—(I) first, as opposed to the existing methods discussed above, the proposed controller ensures global stability, which has been proved using the Lyapunov function. To this end, the sliding surface and corresponding observer variables are selected such that the suitable Lyapunov function can be selected for the global stability analysis; (ii) secondly, instead of using the conventional averaged state-space model for the controller design and analysis, the Brunovsky canonical form of the model for the quadratic converter is derived and used for the controller design. This model accommodates the disturbance variables and aids in the derivation of the control law based on the proposed DO-based sliding surface; (iii) lastly, in order to avoid the chattering and EMI concerns, the PWM method of implementation is used for the implementation of the globally stable SM controller. The main control objective is to regulate the output voltage in the presence of parameter variations such as load changes. An in-depth derivation of the equivalent control law control law and a thorough stability analysis are presented. The suitability of the proposed control scheme has been authenticated by simulation results performed in MATLAB Simulink. It is important to mention that the design methodology of the given control law is such that it can be applied for the control of other types of high-step-up converters as well. The manuscript is structured as follows. In Section 2, the circuit diagram along with scheme has been authenticated by simulation results performed in MATLAB Simulink. It is important to mention that the design methodology of the given control law is such that it can be applied for the control of other types of high-step-up converters as well. The manuscript is structured as follows. In Section 2, the circuit diagram along with an averaged model of the QB converter is given. Next, Section 3 discusses the detailed control law design and the global stability analysis of the system. Finally, in Section 4, some an averaged model of the QB converter is given. Next, Section 3 discusses the detailed control law design and the global stability analysis of the system. Finally, in Section 4, some simulation results are given to establish the ability of the derived control law to handle large signal disturbances, followed by the conclusion in the last section.

2. State-Space Modeling for Quadratic-Ratio Converter

The quadrtaic boost topology's circuit schematic is depicted in Figure 1a. It has an extra step-up arrangement compared to the second-order conventional boost topology. This additional arrangement primarily consists of an additional boost stage but without an The quadrtaic boost topology's circuit schematic is depicted in Figure 1a. It has an extra step-up arrangement compared to the second-order conventional boost topology. This additional arrangement primarily consists of an additional boost stage but without an additonal active switch. The use of a single active switch reduces the converter switching losses. In summary, to increase the gain of the orthodox step-up topology, two boost converters are combined using one active switch to create this converter [25].

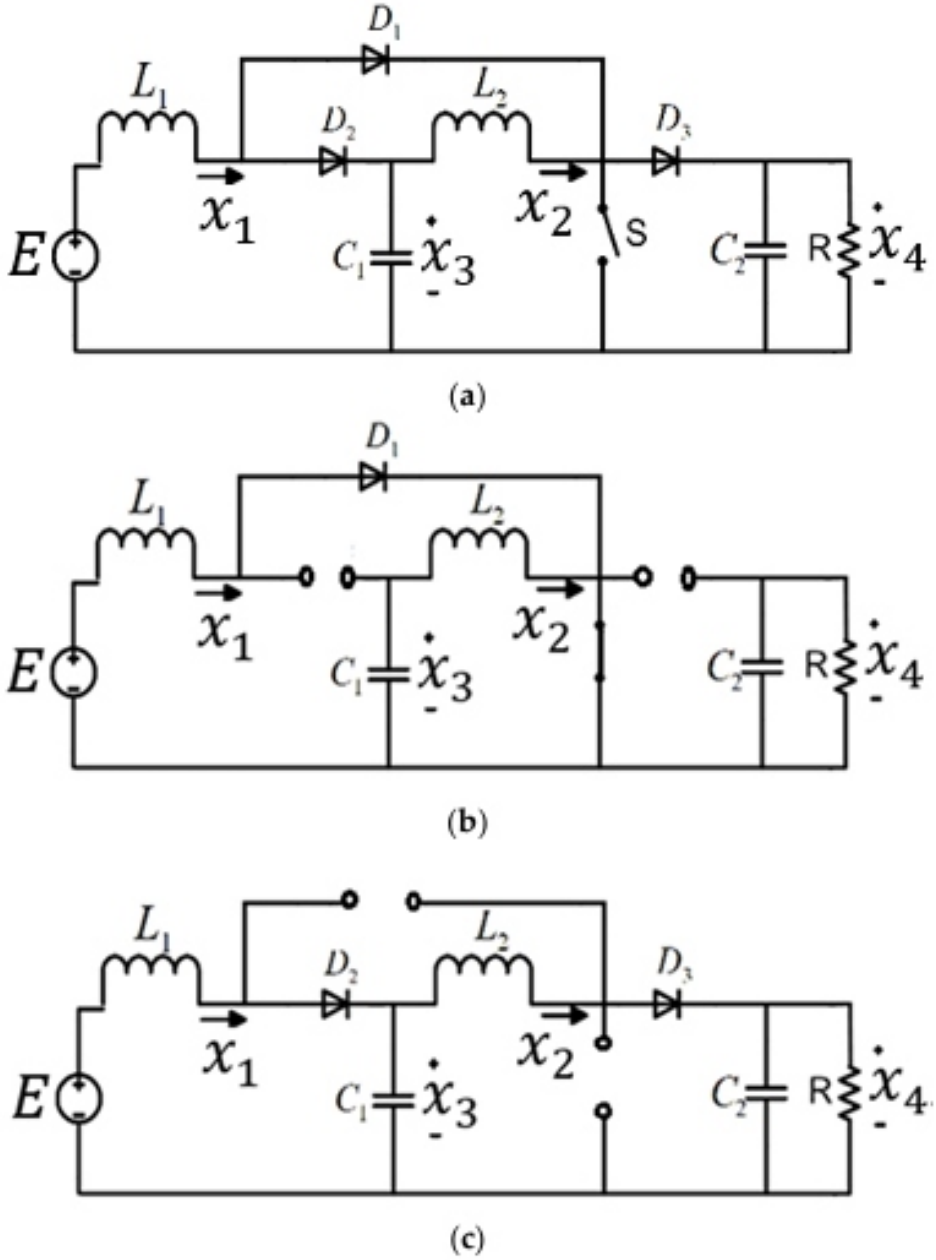


Figure 1. Circuit diagram and operational modes of the quadratic converter: (**a**) Circuit schematic; (**b**) switch ON; (**c**) switch OFF.

averaged modeling equations: (a) The MOSFET 'S' switches on and off in synchrony with all of the diodes; (b) the dc–dc system works in a continuous mode of conduction; (c) all of the diodes and the semiconductor switches are viewed as perfect components with very low parasitic resistance. The following describes the system's two operational modes. 'Mode 1': In this mode, diodes D2 and D3 are biased in the reverse direction while D1 is forward biased. Also, the semiconductor device 'S' is closed while the device is working in this first condition. Energy is stored in the two inductors, L1 and L2, by the input voltage sources E and C1, respectively. The derivative expressions for this mode of operation can be obtained by employing Kirchhoff's laws of voltage and current (KVL and KCL) in Figure 1b, and as a result we obtain (see Appendix A for detailed derivation):

$$\frac{dx_1}{dt} = \frac{E}{L_1} \tag{1}$$

$$\frac{dx_2}{dt} = \frac{x_3}{L_2} \tag{2}$$

$$\frac{dx_3}{dt} = -\frac{x_2}{C_1} \tag{3}$$

$$\frac{dx_4}{dt} = -\frac{x_4}{RC_2} \tag{4}$$

where $x_1 = i_{L_1}$ and $x_2 = i_{L_2}$ are the currents through L_1 and L_2, respectively, and $x_3 = v_{C_1}$ and $x_4 = v_{C_2}$ are the voltages across C_1 and C_2, respectively. Also, E and R are the source voltage and actual load value, respectively.

'Mode 2': In this operational mode, the semiconductor MOSFET 'S' is in an OFF state and D_2 and D_3 are biased in the forward direction while D_1 is also forward biased. This ensures a way for the flow of the inductor current x_1 and x_2 to the load, and the energy from the input and these two inductors is transferred to the load. The derivative expressions for this mode of operation can be obtained by employing Kirchhoff's laws of voltage and current (KVL and KCL) to Figure 1c, for which we obtain (see Appendix B for detailed derivation):

$$\frac{dx_1}{dt} = \frac{E - x_3}{L_1} \tag{5}$$

$$\frac{dx_2}{dt} = \frac{x_3 - x_4}{L_2} \tag{6}$$

$$\frac{dx_3}{dt} = \frac{x_1 - x_2}{C_1} \tag{7}$$

$$\frac{dx_4}{dt} = \frac{x_2}{C_2} - \frac{x_4}{RC_2} \tag{8}$$

Next, the averaged state-space model of the QB converter is obtained. In this technique, the differential equations for the 'ON' state and the 'OFF' state are averaged over one switching period, 'T'. Basically, the 'ON' state equations given by (1)–(4) are multiplied by kT, the 'OFF' state equations given by (5)–(8) are multiplied by $(1-k)T$, and then these equations are added with each other and divided by the total time period, 'T'. Here, k is the duty ratio which is also the control signal of the converter such that $0 < k < 1$. Using (1)–(8), one can obtain the averaged state-space expression of the system, given by:

$$\frac{dx_1}{dt} = -\frac{1}{L_1}(1-k)x_3 + \frac{1}{L_1}E \tag{9}$$

$$\frac{dx_2}{dt} = -\frac{1}{L_2}(1-k)x_4 + \frac{1}{L_1}x_3 \tag{10}$$

$$\frac{dx_3}{dt} = \frac{1}{C_1}(1-k)x_1 - \frac{1}{C_1}x_2 \tag{11}$$

$$\frac{dx_4}{dt} = \frac{1}{C_2}(1-k)x_2 - \frac{1}{RC_2}x_4 \tag{12}$$

Now, one can determine the equilibrium values of the converter by equating (9)–(12) with zero as follows:

$$X_{1\text{ref}} = \frac{V_d^2}{RE},\ X_{2\text{ref}} = \frac{V_d}{R},\ X_{3\text{ref}} = \frac{V_d + E}{2},\ X_{4\text{ref}} = V_d \tag{13}$$

where $X_{1\text{ref}}$, $X_{2\text{ref}}$, $X_{3\text{ref}}$, and $X_{4\text{ref}}$ signify the reference values of x_1, x_2, x_2, and x_4, respectively, and V_d is the reference output voltage.

The aim is to design a suitable non-linear control law to ensure the global stability of this converter when an uncertain load occurs.

3. Controller Design

In this section, on the basis of the averaged model given by (9)–(12), a non-linear SM is designed. The goal is to track the output x_4 to its reference $X_{4\text{ref}}$.

3.1. Transformation into Canonical Form

The widely used averaged state-space model of the dc–dc converter is not suitable to design the proposed controller in order to achieve a large signal stability. To this end, the averaged state-space model given by (9)–(12) is converted into the canonical form such that the first state variable, p_1, is the overall system's energy and the second state variable, p_2, is the difference between the input and output power [26]. This allows the disturbance variables δ_1 and δ_2 to be included in the model and, later, their observers can be used for the design of the SM controller. The model in the revised form is shown below (see Appendix C for detailed derivation):

$$\dot{p_1} = p_2 + \delta_1 \tag{14}$$

$$\dot{p_2} = m + \delta_2 \tag{15}$$

Here, $p_1 = 0.5\left(L_1{x_1}^2 + L_2{x_2}^2 + C_1{x_3}^2 + C_2{x_4}^2\right)$ and $p_2 = Ex_1 - {x_4}^2/R_o$, such that R_o is the system's nominal resistance. Also, the mis-matched disturbance is given by $\delta_1 = {x_4}^2/R_o - {x_4}^2/R$ and the matched disturbance is given by $\delta_2 = (2/R_oC_2)\left(-{x_4}^2/R_o + {x_4}^2/R\right)$. The virtual control law is $m = E^2/L_1 + 2{x_4}^2/R_oC_2 - (Ex_3/L_1 + 2x_2x_4/R_oC_2)(1-k)$.

Next, this model is used for the controller design. From the expression of m, the value of the control signal is obtained as:

$$k = 1 - \frac{C_2{R_o}^2E^2 + 2L_1{v_{C_2}}^2 - L_1C_2{R_o}^2m}{C_2{R_o}^2Ev_{C_1} + 2L_1R_oi_{L_2}v_{C_2}} \tag{16}$$

The original control objective that the actual voltage x_4 follows the reference voltage, $x_{4\text{ref}}$, trajectory is now changed as the state-variables are modified. The new objective is that the state variables p_1 and p_2 follow their reference paths of p_{1ref} and p_{2ref}, respectively, such that:

$$p_{1ref} = 0.5\left(L_1{X_{1ref}}^2 + L_2{X_{2ref}}^2 + C_1{X_{3ref}}^2 + C_2{X_{4ref}}^2\right) \tag{17}$$

$$p_{2ref} = EX_{1ref} - {X_{4ref}}^2/R_o \tag{18}$$

where X_{1ref}, X_{2ref}, X_{3ref}, and X_{4ref} are the reference values of x_1, x_2, x_3, and x_4, respectively. Since EX_{1ref} is the steady-state input power of the converter, and assuming a lossless converter, we can write:

$$P_{ss} = {X_{4ref}}^2 / \bar{R} \tag{20}$$

$$X_{1ref} = \frac{P_{ss}}{E} \tag{21}$$

where P_{ss} is the actual steady-state output power of the converter. Also, it is to be noted that the disturbances δ_1 and δ_2 are assumed to be bounded such that:

$$|\delta_i(t)| \leq \bar{\delta}_i \text{ and } \left|\dot{\delta}_i(t)\right| \leq {\delta_i}^* \tag{22}$$

where $\bar{\delta}_i$ and ${\delta_i}^*$ are positive constants.

3.2. SM Scheme Based on Current through Input Inductor

Initially, an observer to estimate the disturbances in δ_1 and δ_2 is written as [27]:

$$\hat{\delta}_1 = G_{d1} p_1 + \alpha_1 \tag{23}$$

$$\dot{\alpha_1} = -G_{d1}\left(p_2 + \hat{\delta}_1\right) \tag{24}$$

and

$$\hat{\delta}_2 = G_{d2} p_2 + \alpha_2 \tag{25}$$

$$\dot{\alpha_2} = -G_{d2}\left(m + \hat{\delta}_2\right) \tag{26}$$

where G_{d1} and G_{d2} are the constant gains of an observer and α_1 and α_2 are the auxiliary gains of an observer. Using (23)–(26), we obtain:

$$\dot{e}_{\delta 1} = -G_{d1} e_{\delta 1} + \dot{\delta}_1 \tag{27}$$

Also:

$$\dot{e}_{\delta 2} = -G_{d2} e_{\delta 2} + \dot{\delta}_2 \tag{28}$$

where $e_{\delta 1} = \delta_1 - \hat{\delta}_1$ and $e_{\delta 2} = \delta_2 - \hat{\delta}_2$ are the errors in the estimation of two disturbances. Next, considering the difficulty of measuring the actual output power, its estimation is obtained as:

$$\hat{P}_{ss} = \frac{{X_{4ref}}^2}{R} + \delta_{1ref} - \hat{\delta}_1 \tag{29}$$

Now, $\delta_{1ref} = {X_{4ref}}^2 / R_o - {X_{4ref}}^2 / R$. Thus,

$$\hat{P}_{ss} = \frac{{X_{4ref}}^2}{R_o} - \hat{\delta}_1 \tag{30}$$

Substituting (30) in (17) and (19), and using (21), the estimation of the reference values of new state variables is obtained using:

$$\hat{p}_{1ref} = \frac{1}{2}\frac{L_1}{E^2}\left(\frac{{X_{4ref}}^2}{R_o} - \delta_1\right)^2 + \frac{1}{2}C_1 {X_{3ref}}^2 + \frac{1}{2}L_2 {X_{2ref}}^2 + \frac{1}{2}C_2 {X_{4ref}}^2 \tag{31}$$

$$\hat{p}_{2ref} = -\hat{\delta}_1 \tag{32}$$

where $\hat{p}_{1ref}$ is the estimation of p_{1ref} and $\hat{p}_{2ref}$ is the estimation of p_{2ref}.

Next, the form of the proposed SM controller for the regulation of the QB converter is proposed as:

$$s = ce_{p_1} + e_{p_2} - \dot{\hat{p}}_{1ref} \tag{33}$$

where $e_{p_1} = p_1 - \hat{p}_{1ref}$ and $e_{p_2} = p_2 - \hat{p}_{2ref}$ are the errors in the new state variables p_1 and p_2, respectively, and c is the constant gain. In order to satisfy the condition that 's' converges to zero, we obtain:

$$m = -c\left(e_{p_2} - \dot{\hat{p}}_{1ref}\right) + \ddot{\hat{p}}_{1ref} - \dot{\hat{\delta}}_1 - \hat{\delta}_2 - K_{b1}sgn(s) - K_{b2}s \tag{34}$$

where K_{b1} and K_{b2} are user-defined controller gains.

3.3. Global Stability Analysis

Next, the Lyapunov function is selected such that $V(s)$ is positive definite and the condition $\dot{V}(s) \leq 0$ can be satisfied for certain values of the controller gains. We need $\dot{V}(s)$ to be negative definite to prove the asymptotic stability. Thus, let us define the Lyapunov function as given by (35).

$$V(s) = \frac{1}{2}s^2 \tag{35}$$

Using (14) and (15), (27) and (28), and (32)–(34), we obtain:

$$\dot{s} = ce_{\delta_1} + e_{\delta_2} - K_{b1}sgn(s) - K_{b2}s \tag{36}$$

Using (35) and (36), we obtain:

$$\dot{V}(s) = s\dot{s} = s\left(ce_{\delta_1} + e_{\delta_2} - K_{b1}sgn(s) - K_{b2}s\right) \tag{37}$$

Since $s.sgn(s) = |s|$, we obtain:

$$\dot{V}(s) = -K_{b1}|s| - K_{b2}s^2 + s\left(ce_{\delta_1} + e_{\delta_2}\right) \tag{38}$$

Thus:

$$\dot{V}(s) \leq -K_{b1}|s| - K_{b2}s^2 + |s|\left(ce_{\delta_{1max}} + e_{\delta_{2max}}\right) \tag{39}$$

where $e_{d_{imax}} = sup\left|e_{\delta_i}(t)\right|, t > 0$ and $i \,\epsilon\{1,2\}$. From (35), $|s| = \sqrt{2}V^{\frac{1}{2}}$. Thus, (39) becomes:

$$\dot{V}(s) \leq -\sqrt{2}V^{\frac{1}{2}}\left[K_{b1} + K_{b2}|s| - \left(ce_{\delta_{1max}} + e_{\delta_{2max}}\right)\right] \tag{40}$$

Now, $\dot{V}(s) \leq 0$, as long as $(K_{b1} + K_{b2}|s|) > (ce_{\gamma_{1max}} + e_{\gamma_{2max}})$.

Thus, the QB converter controlled by the proposed SM controller is asymptotically stable even when large parameter variations occur, if the controller gains are selected such that $(K_{b1} + K_{b2}|s|) > \left(ce_{\delta_{1max}} + e_{\delta_{2max}}\right)$. Next, using $s = 0$ in (33), we obtain:

$$e_{p_2} = -ce_{p_1} + \dot{\hat{p}}_{1ref} \tag{41}$$

Thus, using the definitions of e_{p_1}, e_{p_2}, and $e_{\delta 1}$, and using (14), (32), and (41), we obtain:

$$\dot{e}_{p_1} = -ce_{p_1} + e_{\delta 1} \tag{42}$$

Finally, the dynamics of the controlled system are stated by:

$$\dot{e} = Me + N\dot{\delta} \tag{43}$$

where $\dot{e} = \left[\dot{e}_{p_1}\ \dot{e}_{\delta 1}\ \dot{e}_{\delta 2}\right]$ and $\dot{\delta} = \left[\dot{\delta}_1\ \dot{\delta}_2\right]$, and the matrices M and N, are given by:

$$M = \begin{bmatrix} -c & 1 & 0 \\ 0 & -G_{d1} & 0 \\ 0 & 0 & -G_{d2} \end{bmatrix} \text{ and } N = \begin{bmatrix} 0 & 0 \\ 1 & 0 \\ 0 & 1 \end{bmatrix}$$

Here, M is the stability matrix of the system. A square matrix, M, is called a Hurwitz matrix if all eigenvalues of M have strictly negative real parts, i.e., Re[λi] < 0, where λi is the ith eigen value. Since all of the constants in the matrix M viz. c, G_{d1}, and G_{d2} are positive constants and their coefficients are negative, the matrix M is a Hurwitz matrix. The system is globally stable [28]. Also, since there exists an input-to-state system of the form $\dot{p} = f(p, \delta)$ which is globally stable and also the input of which satisfies the condition $\lim_{t\to\infty} \delta(t) = 0$, the system states satisfy the condition $\lim_{t\to\infty} p(t) = 0$. Thus, the system errors asymptotically converge such that $\lim_{t\to\infty} e_{p_1}(t) = 0$, $\lim_{t\to\infty} e_{\delta 1}(t) = 0$, and $\lim_{t\to\infty} e_{\delta 1}(t) = 0$.

Discussions: The key element in addressing the control of the proposed system is the eigenproblem. All negative coefficients in the stability matrix of the system dynamics indicate that all of the real parts of the eigen values are in the left half of the plane. This proves the bounded nature of the system as time approaches infinity. Such an eigenproblem has been previously used to validate the stability of several control systems, as described in [29–32].

4. Simulation Outcomes

In this section, some simulation outcomes are presented to validate the use of the proposed control scheme to regulate the QB converter. The control scheme was realized in MATLAB Simulink version 2022b. Figure 2 shows the block diagram of the control scheme's realization. The converter parameters used were: $E = 10$ V, $L_1 = 180$ uH, $L_2 = 180$ uH, $C_1 = 930$ uF, $C_2 = 930$ uF, $R = 100\ \Omega$, and $X_{4ref} = 40$ V. Also, the controller gains used were: $G_{d1} = 100$, $G_{d2} = 100$, $c = 8000$, $K_{b1} = 2000$, and $K_{b2} = 500$. Next, it is worth mentioning that the proposed controller is based on the pulse width modulation-based approach in which the control signal k is compared with a carrier sawtooth signal of fixed frequency and the resulting PWM signal is generated using a comparator. The switching frequency used was 100 KHz. The output PWM signal of a comparator then drives the switch of the quadratic boost converter. The control input is the duty signal given by (16).

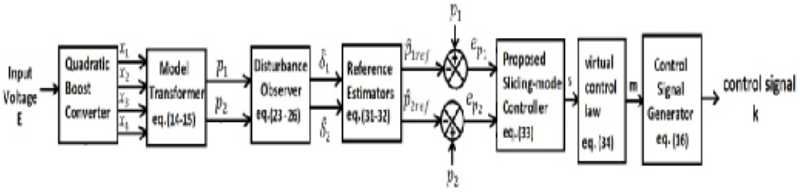
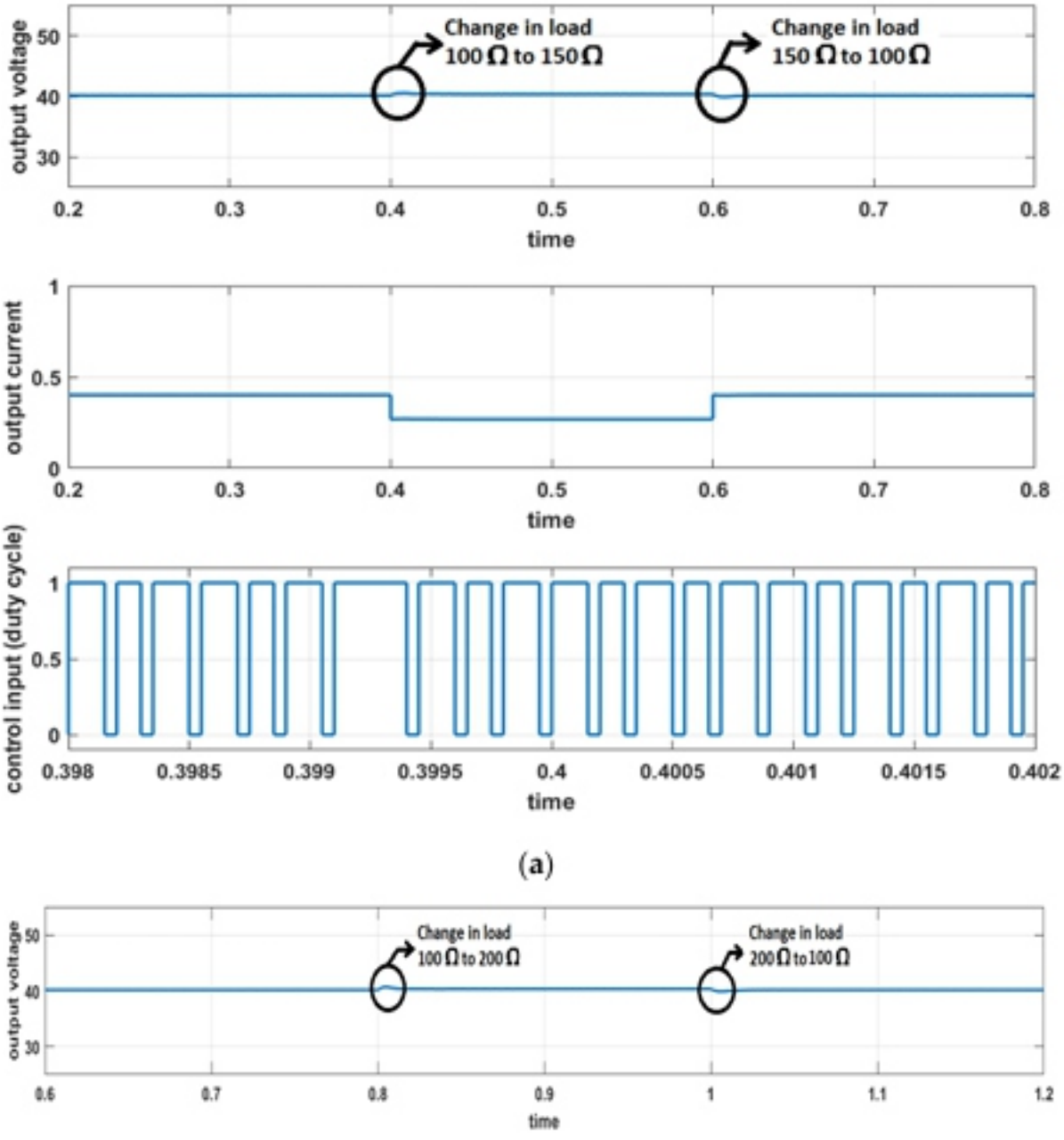
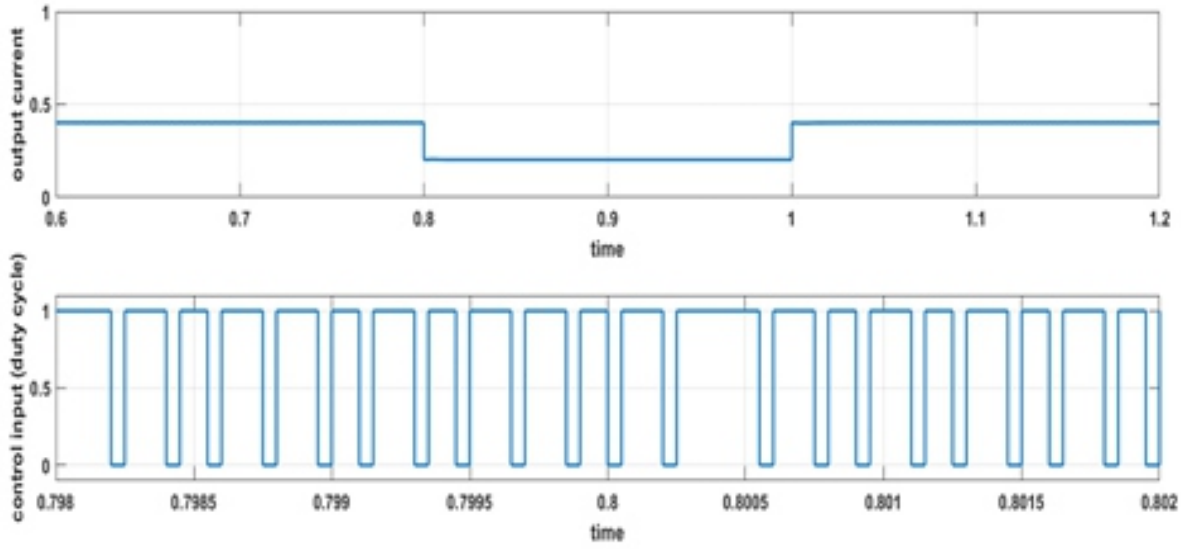


Figure 2. Block diagram of the control scheme's realization.

Initially, the effect of load disturbances on the output response was investigated. Figure 3a shows the system's response when the load was varied by 50% from $R = 100\ \Omega$ to $R = 150\ \Omega$ at time t = 0.4 s, and then back to $R = 100\ \Omega$ at t = 0.6 s. The response in Figure 3a includes the response of the output voltage, the load current change, and a zoomed version of the control signal. It can be observed that the output quickly reached the reference voltage. The response has an overshoot of ~1% and settling time of ~0.01 s. The ability of the converter to handle heavy load disturbances was also investigated and, to this end, the load was changed from 100% of its nominal value. Figure 3b shows the system's response when the load was varied by 100% from $R = 100\ \Omega$ to $R = 200\ \Omega$ at time t = 0.8 s, and then back to $R = 100\ \Omega$ at t = 1 s. Again, the output settled to its reference value with an overshoot of only ~1.2% and a settling time of ~0.02 s.

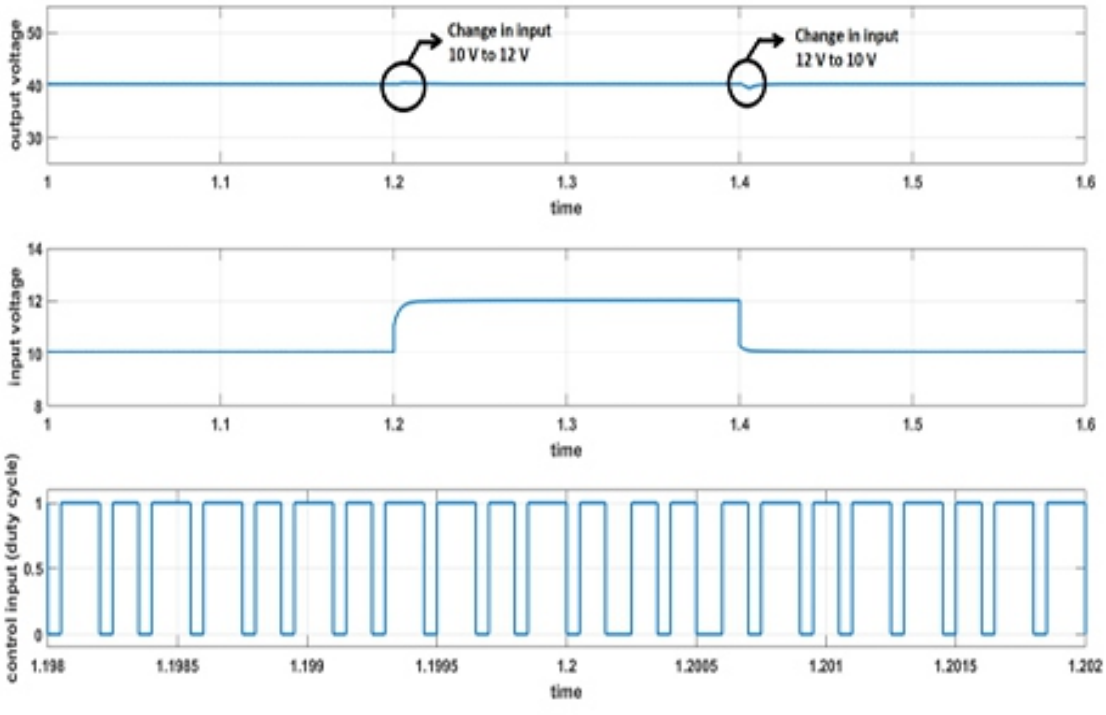




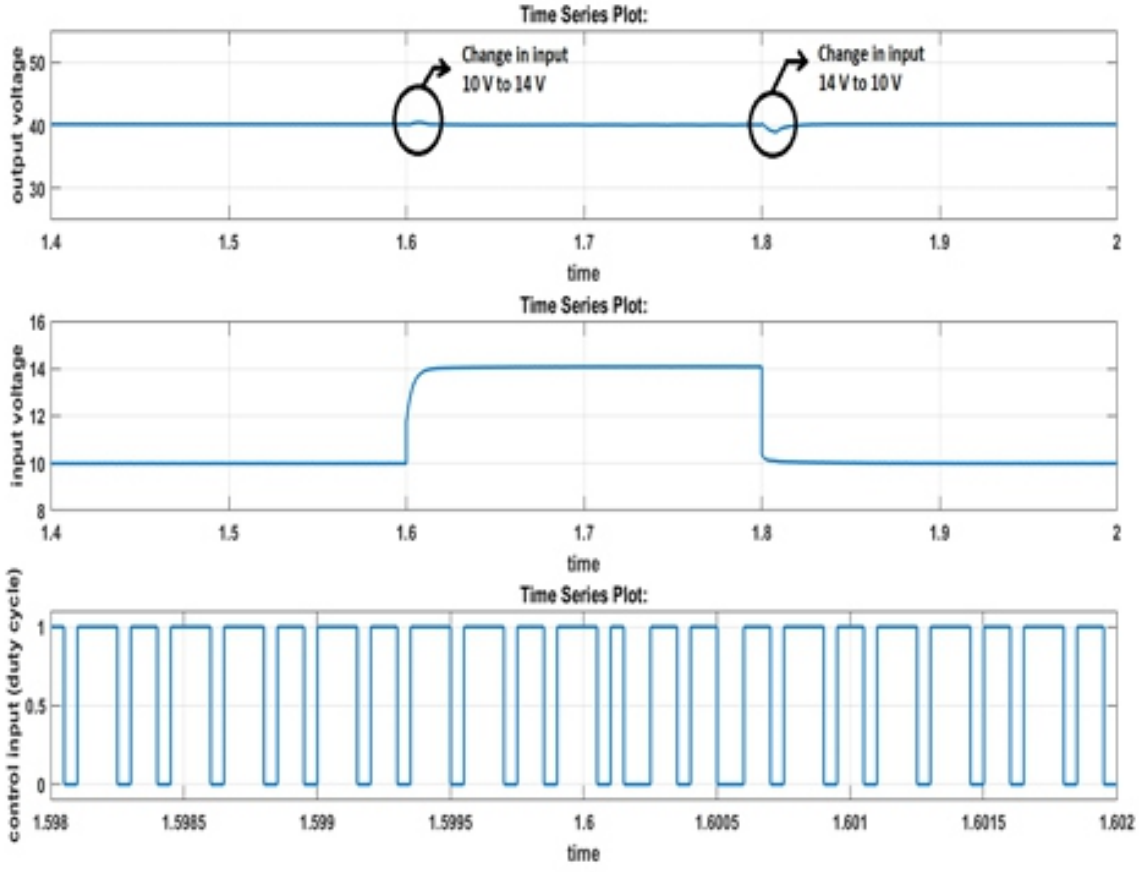
(**b**)

Figure 3. Load-change response (output voltage response: middle; output current response: center; zoomed control signal: bottom). (**a**) 50% variation in load from $R = 100\ \Omega$ to $R = 150\ \Omega$ at time t = 0.4 s, and then back to $R = 100\ \Omega$ at t = 0.6 s; (**b**) 100% variation in load from $R = 100\ \Omega$ to $R = 200\ \Omega$ at time t = 0.8 s, and then back to $R = 100\ \Omega$ at t = 1 s.

Next, the ability of the closed-loop system to handle the input voltage variations was verified. First, the input battery voltage was changed by 20% from $E = 10$ V to $E = 12$ V at t = 1.2 s, and then back to $E = 10$ V at t = 1.4 s. Figure 4a shows the converter's response and the corresponding input voltage and control signal variables. Next, the supply was changed by 50% from 10 V to 14 V at t = 1.6 s and then back to 10 V at t = 1.8 s. The response is depicted in Figure 4b. As can be observed, the response settles to its nominal value in ~0.02 s.



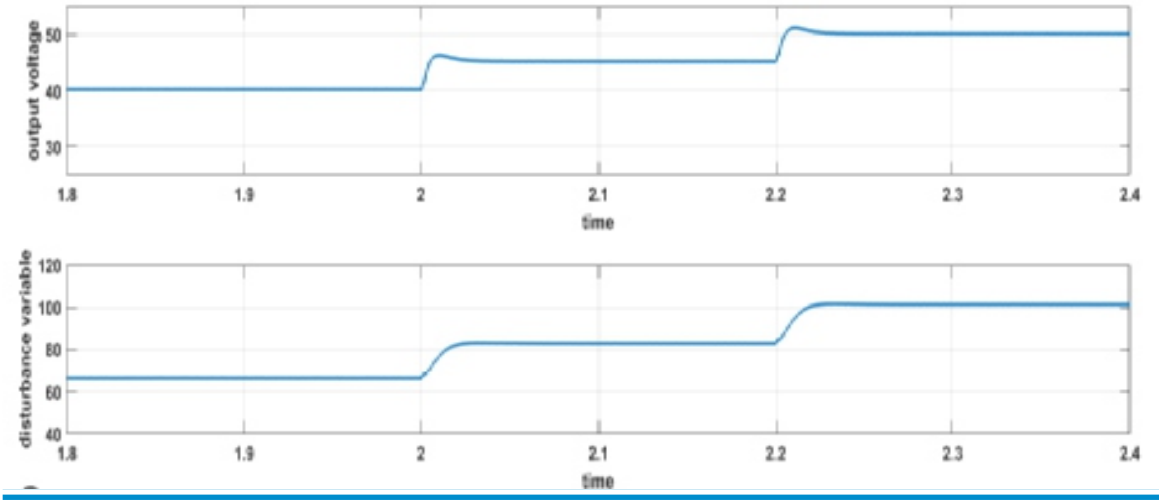
(**a**)



(**b**)

Figure 4. Line change response (output voltage response: middle; input voltage response: center; zoomed control signal: bottom). (**a**) 20% change in input from $E = 10$ V to $E = 12$ V at t = 1.2 s and then back to $E = 10$ V at t = 1.4 s; (**b**) 50% change in input from 10 V to 14 V at t = 1.6 s and then back to 10 V at t = 1.8 s.

Lastly, the capacity of the propsoed SM-controlled system to handle reference voltage variations was investigated. Figure 5 shows the converter's response, including the output voltage, disturbance variable, and control signal variables when the reference volatge was changed from $X_{4ref} = 40$ V to $X_{4ref} = 45$ V at t = 2 s and then $X_{4ref} = 50$ V at t = 2.2 s. All of these results verify the converter's ability to tightly regulate the output voltage to its reference value.



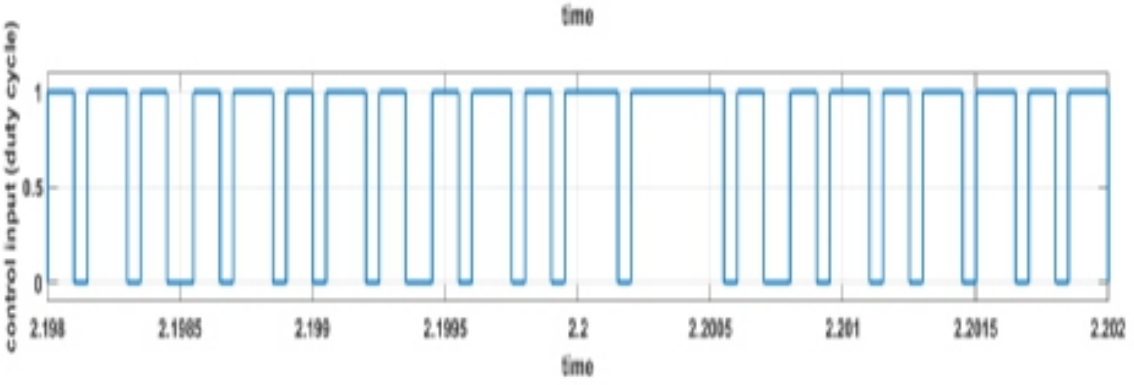


Figure 5. Reference voltage change response of the system when reference volatge was changed from $X_{4ref} = 40$ V to $X_{4ref} = 45$ V at t = 2 s and then $X_{4ref} = 50$ V at t = 2.2 s.

5. Conclusions

In this article, a detailed design and analysis of a globally stable SM controller for the high-step-up quadratic boost converter is presented. The detailed controller design, including the derivation of the control signal, is presented. The proposed controller employs observer variables of the disturbances which estimate the changes in the system's power and capacitor current. The main contribution of this paper is that the Lyapunov stability criterion is employed to validate the large signal stability of the SM-controlled QB converter. Also, the PWM-based SM control scheme has been employed to avoid the chattering effect. Finally, some simulation outcomes are shown to support the theoretical results. They validate the ability of the proposed controller to handle the load, line, and reference voltage variations. It is important to note that, while the suggested controller is used to control a quadratic boost converter, it can also easily be implanted in other high-order dc–dc converters to control their output voltage.

Author Contributions: Conceptualization, S.C.; Formal analysis, S.C.; Funding acquisition, S.U. and M.T.; Investigation, S.C.; Methodology, S.C.; Validation, S.C.; Writing—original draft, S.C.; Writing—review & editing, S.C. and M.T.; Supervision, S.U. and M.T.; Project Administration, S.U. and M.T. All authors have read and agreed to the published version of the manuscript.

Funding: This research was funded by Princess Nourah bint Abdulrahman University Researchers Supporting Project Number (PNURSP2023R79), Princess Nourah bint Abdulrahman University, Riyadh, Saudi Arabia.

Data Availability Statement: Not applicable.

Acknowledgments: Princess Nourah bint Abdulrahman University Researchers Supporting Project Number (PNURSP2023R79), Princess Nourah bint Abdulrahman University, Riyadh, Saudi Arabia.

Conflicts of Interest: The authors declare no conflict of interest.

Appendix A

Here, the proofs of Equations (1)–(4) are presented for the 'ON' state of the QB converter. In Figure 1b, applying KVL in the loop, including E, L_1, D_1, and closed switch S, we obtain:

$$E - v_{L1} = E - L_1 \frac{dx_1}{dt} = 0 \Rightarrow \frac{dx_1}{dt} = \frac{E}{L_1}. \quad \text{(A1)}$$

where v_{L1} is the voltage across inductor L_1, which is equal to $L_1 \frac{dx_1}{dt}$.

Similarly, applying KVL in the loop, including C_1, L_2, and closed switch S, we obtain:

$$x_3 - v_{L2} = x_3 - L_2 \frac{dx_2}{dt} = 0 \Rightarrow \frac{dx_2}{dt} = \frac{x_3}{L_2}. \quad \text{(A2)}$$

where v_{L2} is the voltage across inductor L_2, which is equal to $L_2\frac{dx_2}{dt}$.

Next, if the current through capacitor C_1 is assumed as i_{C1} which is equal to $C_1\frac{dx_3}{dt}$, then:

$$i_{C1} = -x_2 \Rightarrow C_1\frac{dx_3}{dt} = -x_2 \Rightarrow \frac{dx_3}{dt} = -\frac{x_2}{C_1}. \tag{A3}$$

Similarly, if the current through capacitor C_2 is assumed as i_{C2}, which is equal to $C_o\frac{dx_4}{dt}$, then:

$$i_{C2} = -\frac{x_4}{R} \Rightarrow C_2\frac{dx_4}{dt} = -\frac{x_4}{R} \Rightarrow \frac{dx_4}{dt} = -\frac{x_4}{RC_2}. \tag{A4}$$

Equations (A1)–(A4) prove Equations (1)–(4), respectively.

Appendix B

Here, the proofs of Equations (5)–(8) are presented for the 'OFF' state of the QB converter. In Figure 1c, applying KVL in the loop, including E, L_1, and C_1, we obtain:

$$E - v_{L1} - x_3 = 0 \Rightarrow E - L_1\frac{dx_1}{dt} - x_3 = 0 \Rightarrow \frac{dx_1}{dt} = \frac{E - x_3}{L_1}. \tag{A5}$$

Also, applying KVL in the loop including C_1, L_2, and C_2, we obtain:

$$x_3 - v_{L2} - x_4 = x_3 - L_2\frac{dx_2}{dt} - x_4 = 0 \Rightarrow \frac{dx_2}{dt} = \frac{x_3 - x_4}{L_2}. \tag{A6}$$

Next, if the current through capacitor C_1 is assumed as i_{C1}, which is equal to $C_1\frac{dx_3}{dt}$, then:

$$i_{C1} = x_1 - x_2 \Rightarrow C_1\frac{dx_3}{dt} = x_1 - x_2 \Rightarrow \frac{dx_3}{dt} = \frac{dx_3}{dt} = \frac{x_1 - x_2}{C_1}. \tag{A7}$$

Similarly, if the current through capacitor C_2 is assumed as i_{C2}, which is equal to $C_o\frac{dx_4}{dt}$, then:

$$i_{C2} = x_2 - \frac{x_4}{R} \Rightarrow C_2\frac{dx_4}{dt} = x_2 - \frac{x_4}{R} \Rightarrow \frac{dx_4}{dt} = \frac{x_2}{C_2} - \frac{x_4}{RC_2}. \tag{A8}$$

Equations (A5)–(A8) prove Equations (5)–(8), respectively.

Appendix C

The transformed system's first state variable is given as:

$$p_1 = 0.5\left(L_1{x_1}^2 + L_2{x_2}^2 + C_1{x_3}^2 + C_2{x_4}^2\right) \tag{A9}$$

Taking first derivative of (A9), we obtain:

$$\dot{p}_1 = L_1x_1\dot{x}_1 + L_2x_2\dot{x}_2 + C_1x_3\dot{x}_3 + C_2x_4\dot{x}_4 \tag{A10}$$

Substituting (1)–(4) in (A10), we obtain:

$$\dot{p}_1 = x_1(-(1-k)x_3 + E) + x_2(-(1-k)x_4 + x_3) + x_3((1-k)x_1 - x_2) + x_4\left((1-k)x_2 - \frac{1}{\mathrm{R}}x_4\right) \tag{A11}$$

Simplifying (A11), we obtain:

$$\dot{p}_1 = Ex_1 - \frac{{x_4}^2}{R} \tag{A12}$$

Using the transformed system's second state variable as $p_2 = Ex_1 - {x_4}^2/R_o$, we obtain:

$$\dot{p}_1 = p_2 + \frac{{x_4}^2}{R_o} - \frac{{x_4}^2}{R}. \tag{A13}$$

Now, if disturbance variable $\delta_1 = \frac{{x_4}^2}{R_o} - \frac{{x_4}^2}{R}$, we obtain:

$$\dot{p}_1 = p_2 + \delta_1 \tag{A14}$$

Next, from $p_2 = Ex_1 - {x_4}^2/R_o$ as defined in (A12), we obtain:

$$\dot{p}_2 = \mathrm{E}\dot{x}_1 - 2\frac{x_4}{R_o}\dot{x}_4 \tag{A15}$$

Substituting (1) and (4) in (A15) we obtain:

$$\dot{p}_2 = \mathrm{E}\left(-\frac{1}{L_1}(1-k)x_3 + \frac{1}{L_1}E\right) - 2\frac{x_4}{R_o}\left(\frac{1}{C_2}(1-k)x_2 - \frac{1}{RC_2}x_4\right). \tag{A16}$$

Solving (A16), we obtain:

$$\dot{p_2} = m + \delta_2 \tag{A17}$$

where $m = E^2/L_1 + 2{x_4}^2/R_oC_2 - (Ex_3/L_1 + 2x_2x_4/R_oC_2)(1-k)$ and $\delta_2 = (2/R_oC_2)$ $\left(-{x_4}^2/R_o + {x_4}^2/R\right)$.

References

1. *Novaes, Y.R.; Barbi, I.; Rufer, A. A new three-level quadratic (T-LQ) DC-DC converter suitable for fuel cell applications. IEEJ Trans. Ind. Appl. 2008, 128, 459–467. [CrossRef]*

2. *Chakraborty, S.; Vu, H.-N.; Hasan, M.M.; Tran, D.-D.; Baghdadi, M.E.; Hegazy, O. DC-DC Converter Topologies for Electric Vehicles, Plug-in Hybrid Electric Vehicles and Fast Charging Stations: State of the Art and Future Trends. Energies 2019, 12, 1569. [CrossRef]*

3. *Ribeiro, E.F.F.; Cardoso, A.M.; Boccaletti, C.; Mendes, A.M.S. Photovoltaic DC-DC converter for Telecommunications Energy Systems. In Proceedings of the 2009 International Conference on Clean Electrical Power, Capri, Italy, 9–11 June 2009. [CrossRef]*

4. *Lee, S.; Kim, P.; Choi, S. High step-up soft-switched converters using voltage multiplier cells. IEEE Trans. Power Electron. 2013, 28, 3379–3387. [CrossRef]*

5. *Luo, F.L. Positive output Luo converters: Voltage lift technique. IEE Proc. Electr. Power Appl. 1999, 146, 415–432. [CrossRef]*

6. *Lopez-Santos, O.; Martinez-Salamero, L.; Garcia, G.; Valderrama-Blavi, H.; Sierra-*

-Polanco, T. Robust sliding-mode control design for a voltage regulated quadratic boost converter. IEEE Trans. Power Electron. 2015, 30, 2313–2327. [CrossRef]

7. Chincholkar, S. H.; Malge, S. V.; Patil, S. L. Design and Analysis of a Voltage-Mode Non-Linear Control of a Non-Minimum-Phase Positive Output Elementary Luo Converter. Electronics 2022, 11, 207. [CrossRef]

8. Chan, C.Y. Comparative study of current-mode controllers for a high-order boost dc–dc converter. IET Power Electron. 2014, 7, 237–243. [CrossRef]

9. Morales-Saldaña, J.A.; Galarza-Quirino, R.; Leyva-Ramos, J.; Carbajal-Gutierrez, E.E.; Ortiz-Lopez, M.G. Multiloop controller design for a quadratic boost converter. IET Electr. Power Appl. 2015, 1, 362–367. [CrossRef]

10. Chincholkar, S.; Tariq, M.; Abdelhaq, M.; Alsaqour, R. Design and Selection of Inductor Current Feedback for the Sliding-Mode Controlled Hybrid Boost Converter. Information 2023, 14, 443. [CrossRef]

11. Jiang, W.; Chincholkar, S.H.; Chan, C.-Y. Investigation of a voltage-mode controller for a DC-DC multilevel boost converter. IEEE Trans. Circuits Syst. II Express Briefs 2018, 65, 908–912. [CrossRef]

12. Dupont, F.H.; Rech, C.; Gules, R.; Pinheiro, J.R. Reduced-order model and control approach for the boost converter with a voltage multiplier cell. IEEE Trans. Power Electron 2013, 28, 3395–3404. [CrossRef]

13. Chincholkar, S.H.; Chan, C. Investigation of current-mode controlled Cascade Boost converter systems: Dynamics and stability issues. IET Power Electron. 2016, 9, 911–920. [CrossRef]

14. Cervantes, I.; Garcia, D.; Noriega, D. Linear Multiloop control of quasi-resonant converters. IEEE Trans. Power Electron. 2003, 18, 1194–1201. [CrossRef]

15. Chan, C.-Y.; Chincholkar, S.H.; Jiang, W. Adaptive current-mode control of a high step-up DC–DC converter. IEEE Trans. Power Electron. 2017, 32, 7297–7305. [CrossRef]

16. He, Y.; Luo, F.L. Sliding-mode control for dc-dc converters with constant switching frequency. IEEE Proc. Control. Theory Appl. 2006, 153, 37–45. [CrossRef]

17. Ravichandran, S.; Patnaik, S.K. Implementation of dual-loop controller for positive output elementary Luo converter. IET Power Electron. 2013, 6, 885–893. [CrossRef]

18. Tan, S.C.; Lai, Y.M. Constant-frequency reduced-state sliding- mode current controller for Cuk Converters. IET Power Electron. 2008, 1, 466–477. [CrossRef]

19. Tan, S.C.; Lai, Y.M.; Tse, C.K. Indirect sliding mode control of power converters via double integral sliding surface. IEEE Trans. Power Electron. 2008, 23, 600–611.

20. Chincholkar, S.H.; Jiang, W.; Chan, C.-Y. A normalized output error-based sliding-mode controller for the DC–DC cascade boost converter. IEEE Trans. Circuits Syst. II Express Briefs 2020, 67, 92–96. [CrossRef]

21. Chincholkar, S.H.; Jiang, W.; Chan, C.-Y. A modified hysteresis-modulation-based sliding mode

control for improved performance in hybrid DC–DC boost converter. IEEE Trans. Circuits Syst. II Express Briefs 2018, 65, 1683–1687. [CrossRef]

22. Chan, C.-Y. Adaptive sliding-mode control of a novel Buck-boost converter based on Zeta Converter. IEEE Trans. Circuits Syst. II Express Briefs 2022, 69, 1307–1311. [CrossRef]

23. Russo, A.; Cavallo, A. Stability and Control for Buck–Boost Converter for Aeronautic Power Management. Energies 2023, 16, 988. [CrossRef]

24. Canciello, G.; Cavallo, A.; Schiavo, A.L.; Russo, A. Multi-objective adaptive sliding manifold control for more electric aircraft. ISA Transactions 2020, 107, 316–328. [CrossRef] [PubMed]

25. Ortiz-Lopez, M.G.; Leyva-Ramos, J.; Carbajal-Gutierrez, E.E.; Morales-Saldana, J.A. Modelling and analysis of Switch-mode Cascade Converters with a single active switch. IET Power Electron. 2008, 1, 478–487. [CrossRef]

26. Sira-Ramirez, H.; Ilic, M. Exact linearization in switched-mode DC-to-DC power converters. Int. J. Control. 1989, 50, 511–524. [CrossRef]

27. Li, S.; Yang, J.; Chen, W.; Chen, X. Disturbance Observer-Based Control: Methods and Applications; CRC Press: Boca Raton, FL, USA, 2017; pp. 73–79.

28. Khalil, H.K.; Grizzle, J. Nonlinear Systems; Prentice Hall: Upper Saddle River, NJ, USA, 2002; pp. 217–222. 29. Yu,J.; Yang, Z.; Kurths, J.; Zhan, M. Small-Signal Stability of Multi-Converter Infeed Power Grids with Symmetry. Symmetry 2021, 13, 157. [CrossRef]

30. Mikhailov, E.; Pashentseva, M. Eigenvalue Problem for a Reduced Dynamo Model in Thick Astrophysical Discs. Mathematics 2023, 11, 3106. [CrossRef]

31. Liu, H.; Li, R.; Ding, Y. Partial Eigenvalue Assignment for Gyroscopic Second-Order Systems with Time Delay. Mathematics 2020, 8, 1235. [CrossRef]

32. Liu, R.; Wang, Z.; Zhang, X.; Ren, J.; Gui, Q. Robust Control for Variable-Order Fractional Interval Systems Subject to Actuator Saturation. Fractal Fract. 2022, 6, 159. [CrossRef]

Instructions for Authors

Essentials for Publishing in this Journal

1 Submitted articles should not have been previously published or be currently under consideration for publication elsewhere.

2 Conference papers may only be submitted if the paper has been completely re-written (taken to mean more than 50%) and the author has cleared any necessary permission with the copyright owner if it has been previously copyrighted.

3 All our articles are refereed through a double-blind process.

4 All authors must declare they have read and agreed to the content of the submitted article and must sign a declaration correspond to the originality of the article.

Submission Process

All articles for this journal must be submitted using our online submissions system. http://enrichedpub.com/ . Please use the Submit Your Article link in the Author Service area.

Manuscript Guidelines

The instructions to authors about the article preparation for publication in the Manuscripts are submitted online, through the e-Ur (Electronic editing) system, developed by **Enriched Publications Pvt. Ltd**. The article should contain the abstract with keywords, introduction, body, conclusion, references and the summary in English language (without heading and subheading enumeration). The article length should not exceed 16 pages of A4 paper format.

Title

The title should be informative. It is in both Journal's and author's best interest to use terms suitable. For indexing and word search. If there are no such terms in the title, the author is strongly advised to add a subtitle. The title should be given in English as well. The titles precede the abstract and the summary in an appropriate language.

Letterhead Title

The letterhead title is given at a top of each page for easier identification of article copies in an Electronic form in particular. It contains the author's surname and first name initial .article title, journal title and collation (year, volume, and issue, first and last page). The journal and article titles can be given in a shortened form.

Author's Name

Full name(s) of author(s) should be used. It is advisable to give the middle initial. Names are given in their original form.

Contact Details

The postal address or the e-mail address of the author (usually of the first one if there are more Authors) is given in the footnote at the bottom of the first page.

Type of Articles

Classification of articles is a duty of the editorial staff and is of special importance. Referees and the members of the editorial staff, or section editors, can propose a category, but the editor-in-chief has the sole responsibility for their classification. Journal articles are classified as follows:

Scientific articles:

1. Original scientific paper (giving the previously unpublished results of the author's own research based on management methods).
2. Survey paper (giving an original, detailed and critical view of a research problem or an area to which the author has made a contribution visible through his self-citation);
3. Short or preliminary communication (original management paper of full format but of a smaller extent or of a preliminary character);
4. Scientific critique or forum (discussion on a particular scientific topic, based exclusively on management argumentation) and commentaries. Exceptionally, in particular areas, a scientific paper in the Journal can be in a form of a monograph or a critical edition of scientific data (historical, archival, lexicographic, bibliographic, data survey, etc.) which were unknown or hardly accessible for scientific research.

Professional articles:

1. Professional paper (contribution offering experience useful for improvement of professional practice but not necessarily based on scientific methods);
2. Informative contribution (editorial, commentary, etc.);
3. Review (of a book, software, case study, scientific event, etc.)

Language

The article should be in English. The grammar and style of the article should be of good quality. The systematized text should be without abbreviations (except standard ones). All measurements must be in SI units. The sequence of formulae is denoted in Arabic numerals in parentheses on the right-hand side.

Abstract and Summary

An abstract is a concise informative presentation of the article content for fast and accurate Evaluation of its relevance. It is both in the Editorial Office's and the author's best interest for an abstract to contain terms often used for indexing and article search. The abstract describes the purpose of the study and the methods, outlines the findings and state the conclusions. A 100- to 250-Word abstract should be placed between the title and the keywords with the body text to follow. Besides an abstract are advised to have a summary in English, at the end of the article, after the Reference list. The summary should be structured and long up to 1/10 of the article length (it is more extensive than the abstract).

Keywords

Keywords are terms or phrases showing adequately the article content for indexing and search purposes. They should be allocated heaving in mind widely accepted international sources (index, dictionary or thesaurus), such as the Web of Science keyword list for science in general. The higher their usage frequency is the better. Up to 10 keywords immediately follow the abstract and the summary, in respective languages.

Acknowledgements

The name and the number of the project or programmed within which the article was realized is given in a separate note at the bottom of the first page together with the name of the institution which financially supported the project or programmed.

Tables and Illustrations

All the captions should be in the original language as well as in English, together with the texts in illustrations if possible. Tables are typed in the same style as the text and are denoted by numerals at the top. Photographs and drawings, placed appropriately in the text, should be clear, precise and suitable for reproduction. Drawings should be created in Word or Corel.

Citation in the Text

Citation in the text must be uniform. When citing references in the text, use the reference number set in square brackets from the Reference list at the end of the article.

Footnotes

Footnotes are given at the bottom of the page with the text they refer to. They can contain less relevant details, additional explanations or used sources (e.g. scientific material, manuals). They cannot replace the cited literature.
The article should be accompanied with a cover letter with the information about the author(s): surname, middle initial, first name, and citizen personal number, rank, title, e-mail address, and affiliation address, home address including municipality, phone number in the office and at home (or a mobile phone number). The cover letter should state the type of the article and tell which illustrations are original and which are not.

Notes: