Network Embedding with Dual Generation Tasks

Jie Liu, Na Li and Zhicheng He
Nankai University, Tianjin, China
jliu@nankai.edu.cn, nali_nku@163.com, hezhicheng@mail.nankai.edu.cn

Abstract
We study the problem of Network Embedding (NE) for content-rich networks. NE models aim to learn efficient low-dimensional dense vectors for network vertices which are crucial to many network analysis tasks. The core problem of content-rich network embedding is to learn and integrate the semantic information conveyed by network structure and node content. In this paper, we propose a general end-to-end model, Dual GEnerative Network Embedding (DGENE), to leverage the complementary information of network structure and content. In this model, each vertex is regarded as an object with two modalities: node identity and textual content. Then we formulate two dual generation tasks. One is Node Identification (NI) which recognizes nodes’ identities given their contents. Inversely, the other one is Content Generation (CG) which generates textual contents given the nodes’ identities. We develop specific Content2Node and Node2Content models for the two tasks. Under the DGENE framework, the two dual models are learned by sharing and integrating intermediate layers, with which they mutually enhance each other. Extensive experimental results show that our model yields a significant performance gain compared to the state-of-the-art NE methods. Moreover, our model has an interesting and useful byproduct, that is, a component of our model can generate texts, which is potentially useful for many tasks.

1 Introduction
Mining content-rich network data arises from many real-world applications. For example, various systems on social platforms often need to cluster users into communities based on users’ following relation and user-generated content. Learning low-dimensional compact representations for network nodes, a.k.a. Network Embedding (NE), plays a very important role for various network analysis problems.

Recently, extensive research efforts have been dedicated to content-rich network embedding. Yang et al. presented text-associated DeepWalk (TADW) to incorporate textual features into NE through matrix factorization [Yang et al., 2015]. To capture deeper content semantics, CANE [Tu et al., 2017] extends LINE [Tang et al., 2015b] with a mutual attention Deep Neural Network (DNN). Other content-rich NE methods include PTE [Tang et al., 2015a] and CENE [Sun et al., 2016].

The critical research issue of content-rich NE is to preserve network structure and node content in the representation learning. But most existing methods fail to capture them adaptively. The reasons are three-fold. (1) The structure-level similarity between vertices can be various. However, most existing NE methods try to preserve designated structure proximities, instead of learning suitable scope of proximity automatically. (2) The fusion strategy for structure and content information is not well studied. Existing methods mostly learn separated structure and content vectors which are combined with naive methods. Actually, they contribute differently from task to task, and it is essential to fuse them adaptively and automatically. (3) Both the structure and content information are highly nonlinear, which makes shallow models ineffective to learn semantic representations. To this end, adaptively learning structure and content preserving deep NE models in a data-driven manner is of great importance.

To address these issues, we propose a novel NE model, Dual GEnerative Network Embedding (DGENE), to learn content-rich node embeddings with two dual cross-modality tasks. In DGENE, each vertex is regarded as an object with two modalities: node identity and textual content. And we formulate two dual generation tasks. One is Node Identification (NI) which recognizes nodes’ identities given their contents. Inversely, the other is Content Generation (CG) which generates textual contents given the nodes’ identities.

To learn flexible order proximity adaptively, we develop novel end-to-end sequence generation models, Content2Node and Node2Content, for the two tasks based on the sequences obtained via random walks. Specifically, Content2Node couples a sequence-to-sequence (seq2seq) model with a CNN for the NI task to read the raw text of each node and predict the corresponding node identity. While in the CG task, we devise a novel hierarchical seq2seq model, Node2Content, which can generate multiple text sequences for the nodes in the input sequence. In addition, enjoying the deep cross-modal encoder-decoder between the content and structure, DGENE is able to integrate the content semantics seamlessly. Under the DGENE framework, the two dual models are learned jointly by sharing and integrating the hidden layers and mutu-
ally regularize each other. As the input and output of one task are exactly the output and input of the other task respectively, NI is naturally the dual task of CG. Such duality reflects the intrinsic complementary relation between Content2Node and Node2Content. Intuitively, learning the dual tasks together will boost the NE performance, as both of them require similar abilities: effective node representations. In the dual learning framework of two cross-modal tasks, the structure and content information can be captured and fused seamlessly.

To summarize, we make the following contributions:

- We propose a dual generative network embedding model that captures both textual contents and network structure. This work is the first attempt to formalize the NE problem as a dual learning task, in which NI and CG are formulated into a unified framework and learned jointly to achieve better performances.
- For each task, we specifically develop novel end-to-end generation models, i.e., a seq2seq model with a CNN for NI task and a stacked seq2seq model for content generation.
- This is the first NE model with the ability of content generation, which makes our DGENE model potentially useful in a wider range of applications.
- Experiments on the tasks of node classification using two real-world datasets demonstrate the superiority of DGENE over various state-of-the-art approaches.

2 Related Work

Early NE works mainly focus on the topology of networks, such as DeepWalk [Perozzi et al., 2014], LINE [Tang et al., 2015b], node2vec [Grover and Leskovec, 2016], GraRep [Cao et al., 2015], and M-NMF [Wang et al., 2017b]. Essentially, these previous approaches mainly focus on the pairwise relation or local structures.

For further improvements, new approaches have been proposed to consider various auxiliary information, like label information [Tu et al., 2016], group information [Chen et al., 2016], network attribute [Wang et al., 2017a], heterogeneous information [Shi et al., 2018], dynamic networks [Yu et al., 2018], and text content [Yang et al., 2015][Sun et al., 2016] [Tang et al., 2015a][Tu et al., 2017]. But these methods usually fail to model the high-order proximities and the nonlinearity of text content. Liu et al. proposed STNE [Liu et al., 2018] to fix these problems, but it considers only the content-to-node translation and biases the embeddings.

Another line of related work is sequence modeling which we exploit to build our model. The adoption of DNN in natural language processing (NLP) has given rise to the use of the recurrent neural network (RNN) [Elman, 1990]. Long short-term memory (LSTM) [Hochreiter and Schmidhuber, 1997], a variant of RNN, has been applied to various tasks like speech recognition [Graves, 2013], sequence tagging [Ma and Hovy, 2016], and classification [Yang et al., 2016]. Moreover, in machine translation [Sutskever et al., 2014], LSTMs are used to both encode and decode sequences, which is called seq2seq. Seq2seq also receives research attention in other NLP tasks like parsing [Vinyals et al., 2015], summarization [Tan et al., 2017], text generation [Li et al., 2015], and multi-task learning [Luong et al., 2015].

Besides, dual learning has been proved to be effective in different tasks. In machine translation, the dual translation processes can benefit from each other through reinforcement learning [He et al., 2016]. Tang et al. built the dual relation between question answering (QA) and question generation (QG) to improve training [Tang et al., 2017]. Xia et al. proposed a general penalty term to strengthen the probabilistic connection between dual supervised learning tasks [Xia et al., 2017]. However, no existing work has utilized dual learning in NE. Hence, we are the first to learn NE via dual tasks and leverage the complementary relations between them.

3 Approach

In a content-rich network $G = (V, E)$, where $V$ and $E$ are the sets of vertices and edges respectively, each vertex $v$ has two modalities, i.e., the node identity $v^i$ and the textual content $v^c$. Generally, the node identity indicates which node it is, while the node content describes what information it conveys. Given a length-$T$ node sequence $S = \{v_1, v_2, \ldots, v_T\}$ sampled by the random walk algorithm, the identity sequence $S^i = \{v^i_1, v^i_2, \ldots, v^i_T\}$ and the corresponding content sequence $S^c = \{v^c_1, v^c_2, \ldots, v^c_T\}$ are a pair of parallel sequences, as defined in [Liu et al., 2018].

To capture long-range proximities and fuse the content and structure information, Liu et al. proposed STNE for the NI task to learn the conditional probability of $S^i$ given $S^c$, i.e., $p(S^i|S^c)$. While in this paper, to better preserve and integrate different modalities, we further define the CG task.

Definition 1: Content Generation. Given a content-rich network, CG is to learn the conditional probability of $S^c$ given $S^i$ for each pair of parallel sequences, i.e., $p(S^c|S^i)$.

The NI and CG tasks have a probabilistic correlation as both tasks relate to the joint probability between $S^i$ and $S^c$. Given $S^i$ and $S^c$, the joint probability $p(S^i, S^c)$ can be computed in two equivalent ways:

$$p(S^i)p(S^c|S^i) = p(S^i)p(S^c|S^i)$$ (1)

The conditional distribution $p(S^c|S^i)$ is exactly the NI model, and $p(S^c|S^i)$ is the CG model. Therefore, we propose a dual generative model to capture the probabilistic correlation between the two tasks and solve them simultaneously.

Figure 1 illustrates the overview of our proposed method. Interestingly, as shown in Figure 1, our DGENE can be regarded as a novel context-aware cross-modal auto-encoder model, since the overall framework of DGENE tries to generate outputs that reconstruct the inputs.

3.1 Content2Node Model for Node Identification

The Content2Node model learns the cross-modal mapping from the content representation space to the identification space, i.e., $p(S^i|S^c; \theta^{NI})$. In other words, it solves the problem of how to identify specific node from its content and structure. Liu et al. proposed a baseline Content2Node model, STNE, which consists of three major components: content embedding, content sequence encoding, and node sequence generation [Liu et al., 2018]. For end-to-end learning
Figure 1: The framework of DGENE. Given a content-rich network, parallel sequences are sampled by the random walk algorithm. Then two dual seq2seq models are jointly learned on them. Finally the intermediate latent representations are adopted as node embeddings.

Figure 2: DGENE for network embedding

### 3.2 Node2Content Model for Content Generation

Inversely, the Node2Content model learns the cross-modal mapping from the node identity space to the content representation space, i.e., $p(S^i | S^c; \theta^{CG})$. In another word, it solves the problem of how to generate text descriptions for nodes according to the structure information. We propose a seq2seq Node2Content model that flexibly integrates the suitable scope of proximities into the cross-modal learning. As illustrated in the right-hand side of Figure 2, Node2Content translates $S^c$ into $S^i$ through node sequence encoding, semantic decoding, and content generation.

#### Node Sequence Encoding

Similar to the content sequence encoding in Content2Node, the node identity sequences $S^i$’s are also encoded with a Bi-LSTM. Prior to that, the embeddings of node identities are obtained by the LookUp$(\cdot, \cdot)$ function:

$$S^i = \text{LookUp}^i(S^c, V) = \{v^i_1, v^i_2, \ldots, v^i_T\},$$

(5)

where $V \in \mathbb{R}^{|V| \times k_v}$ is the embedding matrix for all $|V|$ nodes, $k_v$ is the embedding dimension. And the lookup layer finds out the embedding vector $v^i_j$ for each $v^i_j$ from V.

After the identity embedding sequence $S^i$ is obtained, the Bi-LSTM sequence encoder further encodes it into a context vector $c^{CG}$ according to their structure relation:

$$c^{CG} = \text{Bi-LSTM}(S^i).$$

(6)

#### Semantic Decoding

With the node sequence embedding $c^{CG}$ obtained as above, the semantic decoding step sequentially generates the high-

---

*Proceedings of the Twenty-Eighth International Joint Conference on Artificial Intelligence (IJCAI-19)*)
level semantic representations $d_{CG}^t$s. Each $d_{CG}^t$ has integrated both the identity information of $v_t$ and its structural relation to other nodes in $S$. Here we devise an LSTM layer as the decoder function $D(\cdot, \cdot)$:

$$d_{CG}^t = D(c_{CG}^t, d_{CG}^{t-1}) = \begin{cases} H_{dec}(0, c_{CG}^t) & t = 1 \\ H_{dec}(0, d_{CG}^{t-1}) & t > 1 \end{cases}$$  

(7)

**Content Generation**

Finally, with the decoded semantic representations $d_{CG}^t$s, a text generator is deployed to transform each $d_{CG}^t$ into word sequences, i.e., the textual content of each node. As a conventional practice, an LSTM generator $G(\cdot, \cdot)$ with teacher forcing [Williams and Zipser, 1989] is adopted. Representation of the $l$-th word is generated as:

$$g_{l,t} = G(d_{CG}^t, g_{l,t-1}) = \begin{cases} H_{gen}(0, d_{CG}^t) & l = 1 \\ H_{gen}(g_{l,t-1}, g_{l,t-1}) & l > 1 \end{cases}$$  

(8)

where $u_{l,t-1}$ has different settings in training and generating processes. During training, $u_{l,t-1} = u_{l-1}$ is the embedding of the $(l-1)$-th ground-truth word in $v_c$. While in generation process, it is the embedding of the word $u_{l,t-1}$ predicted at the previous step. The generation process stops when $l$ reaches a predefined maximum length $\gamma$.

With the decoded representation $g_{l,t}$ for the $l$-th word in $v_c$, a fully-connected layer and a softmax layer are utilized to obtain the probabilistic distribution over the whole vocabulary.

$$p_{l,t} = \text{softmax}(FC(g_{l,t}))$$  

(9)

Finally, a cross-entropy layer measures the CG loss:

$$L_{CG}(\theta_{CG}) = - \sum_{v_t \in S} \sum_{l=1}^{u} \sum_{j=1}^{d(v_t,j)} \delta(u_{l,t}, j) p_{l,t}(j)$$  

(10)

where $j$ is the $l$-th ground-truth word in $v_c$.

### 3.3 Learning of the Dual Tasks

In both models, the intermediate layers play the role of connecting the encoder and decoder layers, i.e., $c_{NI}^t$ in Content2Node and $c_{CG}^t$ in Node2Content, which are the pivot points in cross-modal information integration. Moreover, if the dual models can share their intermediate layers in an appropriate manner, they can be tightly coupled effectively and efficiently. Linear combination layers are adopted:

$$\tilde{c}_{NI}^t = FC_{ dual1}(c_{NI}^t + c_{CG}^t, \theta_{Dual1})$$

$$\tilde{c}_{CG}^t = FC_{ dual2}(c_{NI}^t + c_{CG}^t, \theta_{Dual2})$$

(11)

After the sharing and integration process, $\tilde{c}_{NI}^t$ and $\tilde{c}_{CG}^t$ are fed into the decoder of Content2Node and Node2Content respectively so that they can be coupled and learn from each other.

By coupling Content2Node and Node2Content together through parameter sharing in Equation (11), the two models are unified as one loss function:

$$L(\theta) = \tilde{L}_{NI}(\theta_{NI}, \theta_{Dual1}) + \tilde{L}_{CG}(\theta_{CG}, \theta_{Dual1})$$

(12)

where $\tilde{L}_{NI}(\theta_{NI}, \theta_{Dual1})$ and $\tilde{L}_{CG}(\theta_{CG}, \theta_{Dual1})$ are updated by Content2Node and Node2Content models that the inputs to the decoders have been replaced with Equation (11), and $\theta = \{\theta_{NI}, \theta_{CG}, \theta_{Dual1}\}$ is the parameter set.

### 3.4 Node Embedding

Representations in intermediate layers can be taken as node embeddings. In DGENE, hidden representations in the encoders and decoders of Content2Node and Node2Content can be taken as node embeddings, i.e., $x_{NI}(v_t) = [d_{CG}^t; \tilde{h}_{NI}^t; \tilde{h}_{CG}^t]$ and $x_{CG}(v_t) = [d_{CG}^t; \tilde{h}_{CG}^t; \tilde{h}_{CG}^t]$. To fuse information, we concatenate these two embeddings during experiments. It is also worth noting that the node embeddings are context-aware as in CANE [Tu et al., 2017].

The complexity of DGENE is $O(Z \times T \times (M \times k_u \times k_k + \gamma \times k_h \times k_u + \gamma \times k_h \times k_i))$, where $Z, k_k, k_h$ and $k_i$ are the number of node sequences, the kernel size of CNN, the dimension of hidden layers and input features of LSTM in content generation. Compared with other deep learning models, the complexity of DGENE is acceptable.

### 4 Experiments

To investigate the effectiveness of DGENE in modeling both content and structure information in content-rich networks, we compare it with seven NE baselines on two public datasets. Node embeddings are evaluated on classification task. Moreover, model parameters and generated textual contents are also demonstrated to analyze DGENE deeply.

#### 4.1 Datasets

Our DGENE model is evaluated on two real-world scientific paper citation networks. For end-to-end learning purpose, the raw texts of nodes are required.

- **Cora** contains 2211 papers from 7 categories, and there are 5214 citation links between them. Each paper is described by its abstract with an average length of 162. And the vocabulary size is 15,188.

- **Citeseer** contains 4610 papers which are divided into 10 categories. There are 5923 links between these papers. Each paper is described by its title with an average length of 11. And its vocabulary contains 6302 words.

#### 4.2 Comparison Models

To validate the performance of our approach, we compare it against several NE methods:

- **DeepWalk** [Perozzi et al., 2014] uses local information obtained from truncated random walks to learn latent representations by treating them as sentences.

- **LINE** [Tang et al., 2015b] learns large-scale information network embedding using first-order and second-order proximities. We utilize both proximities.

- **GraRep** [Cao et al., 2015] integrates global structure information into node embeddings by matrix factorization.

- **Node2vec** [Grover and Leskovec, 2016] utilizes a biased random walk algorithm to more efficiently explore the neighborhood architecture on the basis of DeepWalk.

- **TADW** [Yang et al., 2015] incorporates text features into network representation by matrix factorization.

- **CANE** [Tu et al., 2017] learns context-aware node embeddings with the mutual attention mechanism, thus can model the semantic relationship between node pairs.
### 4.4 Experimental Setting

For both datasets, we generate $N = 10$ random walks started at each node, and the length of walks is set to $T = 10$. For both encoder and both decoder layers in dual models, we apply dropout with probability $p = 0.2$. For both datasets, we set the matrices $U$ and $V$ randomly initialized, the hidden dimension of encoders $k_0 = 600$, $k_u = 400$, $k_n = 300$, $k_h = 600$, $k_i = 400$, $k_k = [2, 3, 4, 5]$. Besides, we set $M = 100$, $Z = 22, 110$ on Cora dataset and $M = 100$, $Z = 46, 100$ on Citeseer dataset. The dimension of node embeddings is 2400 on both datasets. With a NVIDIA GeForce GTX 1080Ti GPU, the actual running time of an epoch is about 4 minutes on Citeseer dataset and 7 minutes on Cora dataset.

For all compared algorithms, hyper-parameters are set according to the original papers. To eliminate the classifier’s impact on performances, we apply the simple logistic regression classifier. Classification results are evaluated with the micro F1-score. And the percentages of labeled nodes in classification are set to 10%, 30%, 50%, 70%, and 90%.

### 4.4 Node Classification Results

Table 1 and Table 2 demonstrate the classification results on Cora and Citeseer datasets respectively, where the best results among compared models are boldfaced. From these results, we have the following observations and analyses:

- **STNE** [Liu et al., 2018] obtains node embeddings by learning the mapping from content sequences to node sequences with a seq2seq model.

### 4.5 Ablation Analysis

To verify the performance of each component of the model, we conduct the ablation analysis. In both Content2Node and Node2Content, hidden representations in the decoders are taken as node embeddings. Table 3 and Table 4 demonstrate the ablation analysis results on Cora and Citeseer datasets.

It is evident that DGENE performs better than Content2Node and Node2Content. The reason is that the dual learning framework integrates the two opposite translation processes between different modalities. Content2Node performs better than Node2Content on both datasets because the

---

<table>
<thead>
<tr>
<th>Ratio</th>
<th>DeepWalk</th>
<th>LINE</th>
<th>node2vec</th>
<th>GraRep</th>
<th>TADW</th>
<th>CANE</th>
<th>STNE</th>
<th>DGENE</th>
</tr>
</thead>
<tbody>
<tr>
<td>10%</td>
<td>0.740</td>
<td>0.699</td>
<td>0.751</td>
<td>0.727</td>
<td>0.789</td>
<td>0.794</td>
<td>0.799</td>
<td><strong>0.812</strong></td>
</tr>
<tr>
<td>30%</td>
<td>0.815</td>
<td>0.762</td>
<td>0.820</td>
<td>0.804</td>
<td>0.831</td>
<td>0.851</td>
<td>0.833</td>
<td><strong>0.864</strong></td>
</tr>
<tr>
<td>50%</td>
<td>0.816</td>
<td>0.780</td>
<td>0.828</td>
<td>0.817</td>
<td>0.844</td>
<td>0.864</td>
<td>0.852</td>
<td><strong>0.882</strong></td>
</tr>
<tr>
<td>70%</td>
<td>0.827</td>
<td>0.789</td>
<td>0.831</td>
<td>0.822</td>
<td>0.854</td>
<td>0.872</td>
<td>0.855</td>
<td><strong>0.897</strong></td>
</tr>
<tr>
<td>90%</td>
<td>0.829</td>
<td>0.770</td>
<td>0.838</td>
<td>0.811</td>
<td>0.842</td>
<td>0.878</td>
<td>0.856</td>
<td><strong>0.901</strong></td>
</tr>
</tbody>
</table>

**Table 1:** Micro F1-scores on Cora dataset

<table>
<thead>
<tr>
<th>Ratio</th>
<th>DeepWalk</th>
<th>LINE</th>
<th>node2vec</th>
<th>GraRep</th>
<th>TADW</th>
<th>CANE</th>
<th>STNE</th>
<th>DGENE</th>
</tr>
</thead>
<tbody>
<tr>
<td>10%</td>
<td>0.674</td>
<td>0.579</td>
<td>0.683</td>
<td>0.678</td>
<td>0.832</td>
<td>0.781</td>
<td>0.790</td>
<td><strong>0.833</strong></td>
</tr>
<tr>
<td>30%</td>
<td>0.714</td>
<td>0.645</td>
<td>0.724</td>
<td>0.717</td>
<td>0.872</td>
<td>0.804</td>
<td>0.849</td>
<td><strong>0.890</strong></td>
</tr>
<tr>
<td>50%</td>
<td>0.732</td>
<td>0.649</td>
<td>0.737</td>
<td>0.735</td>
<td>0.886</td>
<td>0.820</td>
<td>0.877</td>
<td><strong>0.932</strong></td>
</tr>
<tr>
<td>70%</td>
<td>0.735</td>
<td>0.659</td>
<td>0.756</td>
<td>0.740</td>
<td>0.895</td>
<td>0.823</td>
<td>0.893</td>
<td><strong>0.950</strong></td>
</tr>
<tr>
<td>90%</td>
<td>0.748</td>
<td>0.670</td>
<td>0.772</td>
<td>0.770</td>
<td>0.911</td>
<td>0.857</td>
<td>0.922</td>
<td><strong>0.980</strong></td>
</tr>
</tbody>
</table>

**Table 2:** Micro F1-scores on Citeseer dataset
node content sequences also contain some structure information. Thus Content2Node can be better learned. However, Content2Node can still benefit from Node2Content as their combination, DGENE, further improves over Content2Node. Besides, Content2Node in Table 3, 4 is an extension of STNE with CNN, which performs better than STNE. It illustrates the effectiveness of adopting CNN in Content2Node.

4.6 Parameter Analysis

We evaluate how different values of walk length $T$, walk number $N$ and hidden dimension $k$ affect the performances, while other parameters are fixed. Figure 3 illustrates the analysis about $T$ and $N$. Generally speaking, the F1-scores first rise with the increase of $T$ and $N$ values, and then fall on both datasets. The combination of $T = 10$ and $N = 10$ performs best, which agrees with the above parameter settings.

Figure 4 demonstrates the analysis of the hidden dimension $k$. Regardless of how $k$ changes, the F1-scores rise rapidly in the early stages of training, and gradually reach the stable state, which demonstrates the robustness and stability of DGENE. Among all $k$ values, $k = 300$ achieves the best performances, which conforms to our parameter settings.

4.7 Case Study

As aforementioned, our DGENE is able to generate new node content owing to the Node2Content model. Here we present two examples of generated texts using the DGENE model learned on Cora dataset. With the trained model, a sequence of nodes with contents are fed into it, and we let the decoder of the Node2Content part continue to generate new node content after the content generation for the input nodes.

Figure 5 exhibits two examples of generated new contents (in red boxes) given input content sequences (in blue boxes) about “Markov Chains Monte Carlo” and “genetic algorithm”, respectively. Obviously, the generated texts are semantically coherent with the contextual node contents. It can be concluded that DGENE model can generate text content according to the network context, which implies that meaningful node embeddings are learned in the model. Moreover, this generation ability is potentially useful in many applications, such as automatic web page generation and micro-blog generation, etc.

5 Conclusion

In this paper, we presented DGENE, the first dual learning framework for content-rich network embedding. Specifically, we defined two generation tasks: Node Identification and Content Generation. With the duality, our proposed DGENE leverages the complementary information from the dual tasks, which effectively models the flexible proximity and content semantics in complex networks. Through a joint learning framework, the representations learned by the Node2Content model and the Content2Node model can be mutually enhanced. Moreover, our model is the first NE method that can be applied to generation tasks. Extensive experiments conducted on two real-world datasets demonstrated the effectiveness and superiority of DGENE.

Acknowledgements

This research is supported by the National Natural Science Foundation of China under the grant No. U1633103, the Key Projects in Tianjin Science and Technology Pillar Program under the grant No. 17YFZCGX00610, and the Open Project Foundation of Information Technology Research Base of Civil Aviation Administration of China under the grant No. CAAC-ITRB-201601.
References


