Application of Graph Neural Network in Image Captioning

Hankun Wang
hankun.wang@mail.utoronto.ca
github link: https://github.com/Haokfu/GNN-Encoder-Image-Captioning.git

Abstract

Image captioning is a fundamental vision-to-text task in multimodal learning. Traditional methods employ CNNs to generate a global image embedding, which serves as the initial hidden state for text generation. Inspired by the Vision Transformer (ViT), which introduces patch embeddings as an alternative image representation, this work explores the application of Graph Neural Networks (GNNs) in the image encoder for captioning. Specifically, we model image patches as nodes in a graph, using Graph Convolutional Networks (GCN) and Graph Attention Networks (GAT) to capture spatial relationships through structured feature extraction.

We compare three models: a baseline with a ViT-based encoder, a GCN-based encoder, and a GAT-based encoder. Two key experiments are conducted: (1) a positional encoding study evaluating sinusoidal versus learnable embeddings, and (2) a generalization test where models trained on Flickr8k are evaluated on COCO. Results demonstrate that the GAT encoder with sinusoidal positional encoding achieves the best performance and generalization ability, suggesting that GNN-based encoders can effectively enhance captioning through structured spatial reasoning.

1 Introduction

The image captioning task was introduced in 2015, aiming to train machine learning models to understand images and generate reasonable textual descriptions. Image captioning tasks typically consist of two stages: image understanding and text generation. Traditionally, image understanding is achieved through convolutional neural networks (CNNs) to obtain a latent space image encoding that summarizes the entire image. This encoding is then passed as the initial hidden state to the text generation decoder.

In 2020, with the publication of the Vision Transformer (ViT) (Dosovitskiy et al., 2021), a new solution for image encoding emerged. Instead of summarizing the entire image, the image is split into patches, and encoding is obtained for each patch. encoded image patches are treated as tokens and are passed to the transformer decoder. With the introduction of the attention mechanism in image processing, not only was the vanishing gradient problem efficiently addressed, but the image encoder also gained a more global understanding, which improved the quality of the generated captions. Inspired by the patch encoding solution of Vision Transformer, I propose a new GNN-based architecture for patch embedding in this paper.

Graph Neural Networks (GNNs) can perform node embedding based on the spatial relationships between nodes (Veličković, 2023), which is leveraged in this project to formulate an alternative method of patch embedding.

In the experimental stage of the project, I trained three models: one with a Vision Transformer (ViT) backbone as the baseline image encoder, one with a GCN-based image encoder (Kipf and Welling, 2017), and one with a GAT-based image encoder (Veličković et al., 2018). For text generation, I used a Transformer decoder in all three captioning models (Vaswani et al., 2023). Similar to the ViT approach, the GNN-based encoders also lack positional awareness, so positional encoding were added to the image patches before they were processed by the GNN layers.

Due to computational limitations, model training and hyperparameter tuning were conducted using the small Flickr8k dataset (Hodosh et al.). To evaluate not only the model performance on Flickr8k but also the generalization ability, I tested all three models trained with full Flicker8k dataset on the COCO 2014 validation image captioning dataset (Lin et al., 2015) and used Bleu score as evaluation metrics (Papineni et al., 2002). The results showed that the GAT-based image encoder achieved the best performance among the three models.

2 Related Works

2.1 Image Captioning

Image captioning has evolved significantly in recent years, particularly through the integration of image processing and sequence modeling techniques. In 2015, early models employed CNNs to encode images and LSTMs to generate text (Vinyals et al., 2015), achieving state-of-the-art (SOTA) performance at the time. However, the inherent non-parallelizable nature of recurrent models limited their ability to handle longer captions effectively. This limitation led to the incorporation of attention mechanisms in decoders, which reduced the impact of sequence length and dynamically aligned visual features with relevant words (Xu et al., 2016). Despite these advances, both approaches relied on CNN-based encoders, which represented the entire image with a single latent vector. While CNNs excel at capturing local feature information, they struggle to model global relationships among features, resulting in information loss and local blindness (overemphasis on localized patterns). The introduction of Vision Transformers (ViT) (Dosovitskiy et al., 2021) addressed this issue by splitting images into patch-based tokens processed by transformer encoders, enabling holistic feature modeling. This approach popularized patch embeddings in subsequent captioning tasks. Moreover, the unified transformer architecture for both encoder and decoder enabled large-scale pre-training on unlabeled data before fine-tuning for downstream tasks. Recent efforts have focused on pre-trained vision language models such as BLIP (Li et al., 2022) and GIT (Wang et al., 2022), which use large-scale datasets to improve generalization and fine-grained caption quality.

2.2 GNN for Image Processing

Graph Neural Networks (GNNs) have emerged as a powerful framework for modeling relational data. Early works focused on spectral methods (Bruna et al., 2014), with later advances introducing spatial convolutions (Kipf and Welling, 2017) to enable scalable non-Euclidean data processing. In image processing, GNNs were initially applied to tasks requiring explicit structural modeling, such as scene graph generation (Xu et al., 2017) and point cloud classification (Qi et al., 2017). While traditional CNNs are effective for grid-structured images, they struggle with irregular or hierarchical representations; GNNs address this limitation by treating images as graphs of superpixels (e.g., SLIC-based graphs) or regions (Chen et al., 2018), thereby enabling relational reasoning. Inspired by these advances, I propose to utilize GNN embedding architectures for image patch representations. This approach unifies grid and graph representations of different image patches, facilitating a more informative and globally connected understanding of the image.

3 Methodology

In this project, I developed and trained three models for experimental comparison: ViT Encoder Captioning Model, GCN Encoder Captioning Model, and GAT Encoder Captioning Model. The baseline ViT model adopts the Vision Transformer architecture, but with a reduced hidden dimension due to computational constraints. All three models follow an encoder-decoder framework, where the encoder processes visual inputs (ViT/GCN/GAT) and the decoder generates captions. Detailed descriptions of the encoder and decoder architectures are provided in subsequent sections.

3.1 Encoder Architecture

Among the three encoders, the ViT encoder exhibits a slightly different structure compared to the two GNN-based encoders. In the ViT encoder, the input image is divided into 16 patches. Each patch is then linearly projected into a hidden dimension space and treated as an input token for the subsequent transformer encoder architecture, which generates embedding vectors for all image patches. This process closely follows the standard Vision Transformer (ViT) methodology described in the ViT paper (Dosovitskiy et al., 2021).

The GCN and GAT encoders share nearly identical structures and hidden dimensions. The input image is split into 64 patches, a higher patch count than the 16 patches on the ViT encoder, which improves the quality of the GNN embeddings. Each patch is first projected into the hidden dimension space, with the resulting vector serving as node features for its corresponding image patch. To model spatial relationships, the original image connection information is preserved as edge features in the constructed graph. With both node features and edge connections established, the graph is processed by the graph neural network to generate graph embeddings. Detailed layer architectures are illustrated in Figure 1.

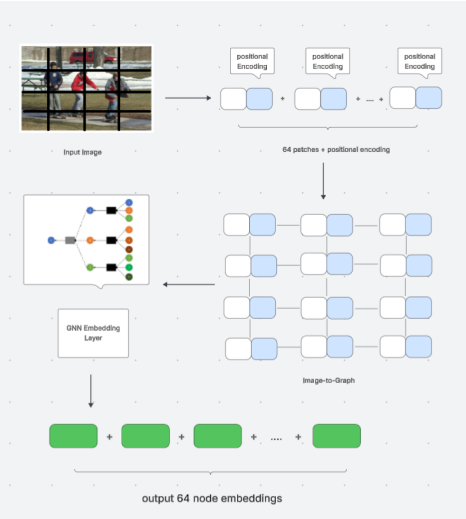

Figure 1: The GNN encoder architecture is illustrated in Figure 1. The input image is divided into 64 patches (simplified to 16 patches in the figure for clarity, though the actual model uses 64). To preserve spatial information, positional encodings are incorporated into the node features. These node features, combined with patch connectivity data, form a graph structure that is processed by GNN layers to generate node embeddings representing the final patch embeddings.

3.2 Decoder architecture

In this project, I employed a single shared decoder model across all three encoder architectures. Since each encoder outputs a sequence of patch embeddings, the decoder is designed to align its input dimensions with the encoder's output shape. These patch embeddings serve as keys and values in the decoder, which follows a standard transformer decoder architecture (albeit with a reduced hidden dimension due to computational constraints). The final output is a probability distribution over the vocabulary. The detailed structure of the decoder is illustrated in Figure 2.

3.3 Training

To train the models, I split the Flickr8K dataset into training, validation, and test sets with a ratio of 6:2:2. Input images are resized to 224×224 pixels and normalized. Target captions are tokenized using the GPT-2 tokenizer and augmented with positional encodings before being fed into the decoder. During training, we evaluate model performance at each epoch by computing the BLEU score on the validation set. Additionally, I employed a learning rate scheduler that reduces the learning rate by a factor of 0.5 if the validation BLEU score fails to improve for two consecutive epochs. The model

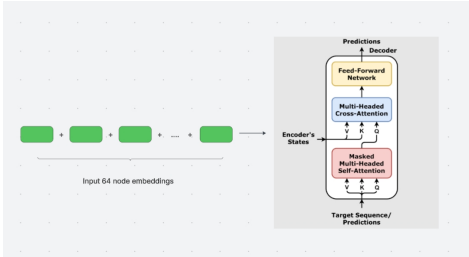

Figure 2: The image encoder outputs a sequence of patch embeddings, each of which is linearly projected to generate key and value representations for the decoder's cross-attention layer. For caption generation, I employed the GPT-2 tokenizer and vocabulary due to its superior performance in text generation tasks.

achieving the highest validation BLEU score during training is saved for final evaluation.

4 Experiment

4.1 Hyperparameter Tuning

To fully demonstrate the capabilities of image captioning, I conducted experiments on hyperparameter tuning for each of the three models using the split Flickr8k dataset. Specifically, Flicker8k dataset I used has 8091 images in total. With ratio of 6:2:2, the splitting number of images for training, validation and test are listed below:

- training set: 4854 images
- validation set: 1618 images
- test set: 1619 images

All three models showed convergence trends toward the end of the training phase. As mentioned earlier, I selected the model that achieved the highest BLEU score during training. However, due to the small size of the validation dataset, there may be some bias in accurately evaluating model performance. Therefore, I also saved the final model at the end of training. For each model, I selected the higher BLEU score on the test set from these two saved models as the final performance score.

4.2 Positional Encoding

To retain the spatial information of the original images, I augmented the patch input features with positional encoding. Two types of positional encoding methods were included in the experiments: sinusoidal positional encoding (Vaswani et al., 2023) and a learnable positional encoding layer.

I trained three models using each positional encoding method (i.e., six models in total) on the Flickr8k training set. To evaluate model performance, the final BLEU score for each model was computed using the BLEU score selection strategy described in the hyperparameter tuning section. Specifically, the flicker8k test set has

4.3 Generalization Test

To evaluate model generalizability, I trained three models on the full Flickr8k dataset using the best hyperparameter sets selected previously, and assessed their BLEU scores on the COCO dataset. Specifically, I used the COCO 2014 Image Captioning Validation dataset, which contains a total of 40,504 images. Since different training datasets may influence the effectiveness of positional encoding methods, I also included models with different positional encoding strategies.

In total, six models were included in the generalization experiment:

- ViT captioning model with sinusoidal positional encoding
- ViT captioning model with learnable positional encoding
- GCN captioning model with sinusoidal positional encoding
- GCN captioning model with learnable positional encoding
- GAT captioning model with sinusoidal positional encoding
- GAT captioning model with learnable positional encoding

During training, since I needed to select the model achieving the highest validation BLEU score as the best saved model and adjust the learning rate accordingly, the validation set was still required. To reduce the bias in dataset scale between hyperparameter tuning and generalization test, I maintained the original training-to-validation ratio and combined the original training and test sets. The final data split for training and validation became 8:2.

5 Results

5.1 Results for Hyperparameter Tuning

The hyperparameter list along with the best choice for each of hyperparameter is shown in table 1.

Hyperparameter	Final Decision
initial learning rate	2e-4
training epoch	20
hidden dimension	128
number of patches	16(ViT), 64(GCN, GAT)
drop out fraction	0.1
batch size	64

Table 1: Except for the number of patches, all other hyperparameters are the same for all three models.

5.2 Results for Positional Encoding

I evaluate three models with different positional encoding methods on flicker8k test dataset. The results for totally six models are illustrated in figure 3.

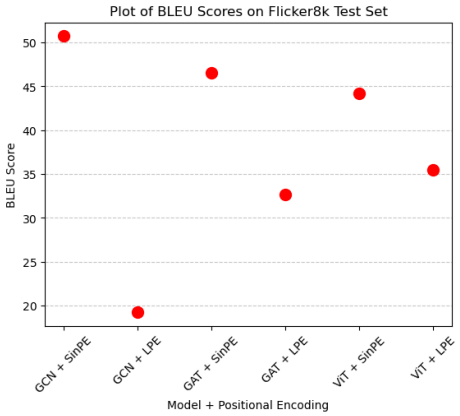

Figure 3: This dot plot shows the BLEU scores of all six models, evaluated on the Flickr8k test dataset.

5.3 Results for Generalization Test

Six models are trained on 80 percent of flicker8k dataset and the best model was selected by its performance on the remaining 20 percent of flicker8k dataset. The results are illustrated in figure 4

6 Analysis

Drawing from the experiments and their corresponding results, we can conduct the following analyses to derive more reasonable conclusions.

6.1 Hyperparameter Tuning

During the hyperparameter tuning process, the number of patches for the ViT model and the GNN-based models differed. Increasing the patch-splitting number significantly improved the performance of the GNN-based models but had only a slight effect on the ViT model. As far as I am

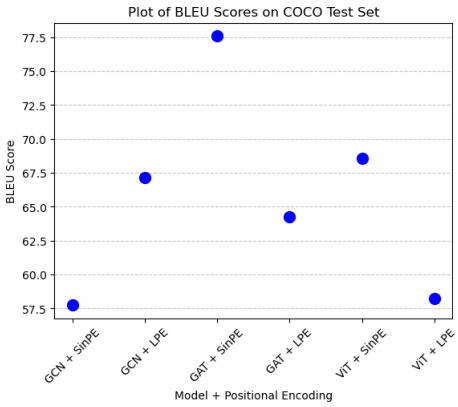

Figure 4: This dot plot shows the BLEU scores of six models, evaluated on COCO dataset

concerned, since GNN layers focus more on the spatial relationships between nodes, having a larger number of nodes allows the encoder to understand image information more effectively.

6.2 Positional Encoding and Model Generalization

From Figure 4, an intuitive analysis suggests that sinusoidal positional encoding performs significantly better than learnable positional encoding. Additionally, the GCN captioning model shows the best performance among the three models. However, the results differ in the generalization experiment. As shown in Figure 5, the GAT captioning model with sinusoidal positional encoding achieves the highest BLEU score. Meanwhile, the GCN captioning model with learnable positional encoding outperforms its counterpart with sinusoidal positional encoding.

7 Conclusion and Future Work

In this project, I primarily explored the application of Graph Neural Networks (GNNs) in the image encoder component of the image captioning task. The two GNN-based architectures used were GCN and GAT. The main innovation of my approach lies in the adaptation of the patch embedding method introduced in the Vision Transformer paper. Instead of treating each patch as a token input, I treated them as nodes and constructed a graph based on the spatial relationships between patches in the input image. By applying GNN embeddings to the patch node representations, the image encoder gains a better understanding of the image, enabling it to provide higher-quality embeddings to the decoder.

With experiments on positional encoding and generalization test, the results indicate that while sinusoidal positional encoding and the GCN architecture perform best on the in-domain Flickr8k dataset, the GAT model with sinusoidal encoding demonstrates superior generalization to out-of-domain data. To conclude, with large amount of training dataset, GAT captioning model should have a better performance and generalization ability.

There are some limitations in this project. I used only a single linear layer to flatten each image patch and generate node features. In future work, I plan to explore alternative mechanisms such as using CNNs or Vision Transformers (ViTs) to extract node features. Another potential approach is to process the entire image with a CNN first and then select node features through receptive field projection.

In addition to the node feature generation layer, the hidden dimension was set to 128 for all encoders and the decoder, due to computational constraints. If more powerful GPUs become available in the future, I would consider increasing the hidden dimension to 512 or 768, and incorporating projection layers between the encoding and hidden dimensions to introduce more representational capacity and flexibility.

References

Joan Bruna, Wojciech Zaremba, Arthur Szlam, and Yann LeCun. 2014. Spectral networks and locally connected networks on graphs.

Yunpeng Chen, Marcus Rohrbach, Zhicheng Yan, Shuicheng Yan, Jiashi Feng, and Yannis Kalantidis. 2018. Graph-based global reasoning networks.

Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, Jakob Uszkoreit, and Neil Houlsby. 2021. An image is worth 16x16 words: Transformers for image recognition at scale.

Micah Hodosh, Peter Young, and Julia Hockenmaier. Flickr8k dataset.

Thomas N. Kipf and Max Welling. 2017. Semi-supervised classification with graph convolutional networks.

Junnan Li, Dongxu Li, Caiming Xiong, and Steven Hoi. 2022. Blip: Bootstrapping language-image pre-training for unified vision-language understanding and generation.

Tsung-Yi Lin, Michael Maire, Serge Belongie, Lubomir Bourdev, Ross Girshick, James Hays, Pietro Perona, Deva Ramanan, C. Lawrence Zitnick, and Piotr Dollár. 2015. Microsoft coco: Common objects in context.

Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. 2002. Bleu: a method for automatic evaluation of machine translation. In *Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics*, pages 311–318, Philadelphia, Pennsylvania, USA. Association for Computational Linguistics.

Charles R. Qi, Hao Su, Kaichun Mo, and Leonidas J. Guibas. 2017. Pointnet: Deep learning on point sets for 3d classification and segmentation.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. 2023. Attention is all you need.

Petar Veličković. 2023. Everything is connected: Graph neural networks. *Current Opinion in Structural Biology*, 79:102538.

Petar Veličković, Guillem Cucurull, Arantxa Casanova, Adriana Romero, Pietro Liò, and Yoshua Bengio. 2018. Graph attention networks.

Oriol Vinyals, Alexander Toshev, Samy Bengio, and Dumitru Erhan. 2015. Show and tell: A neural image caption generator.

Jianfeng Wang, Zhengyuan Yang, Xiaowei Hu, Linjie Li, Kevin Lin, Zhe Gan, Zicheng Liu, Ce Liu, and Lijuan Wang. 2022. Git: A generative image-to-text transformer for vision and language.

Danfei Xu, Yuke Zhu, Christopher B. Choy, and Li Fei-Fei. 2017. Scene graph generation by iterative message passing.

Kelvin Xu, Jimmy Ba, Ryan Kiros, Kyunghyun Cho, Aaron Courville, Ruslan Salakhutdinov, Richard Zemel, and Yoshua Bengio. 2016. Show, attend and tell: Neural image caption generation with visual attention.