

# Exploiting Multiple Sequence Lengths in Fast End to End Training for Image Captioning

Jia Cheng Hu    
*FIM Dept.*  
*Univ. of Modena and Reggio Emilia*  
Modena, Italy  
jiacheng.hu@unimore.it

Roberto Cavicchioli    
*DCE Dept.*  
*Univ. of Modena and Reggio Emilia*  
Reggio Emilia, Italy  
roberto.cavicchioli@unimore.it

Alessandro Capotondi    
*FIM Dept.*  
*Univ. of Modena and Reggio Emilia*  
Modena, Italy  
alessandro.capotondi@unimore.it

**Abstract**—We introduce a method called the Expansion mechanism that processes the input unconstrained by the number of elements in the sequence. By doing so, the model can learn more effectively compared to traditional attention-based approaches. To support this claim, we design a novel architecture ExpansionNet v2 that achieved strong results on the MS COCO 2014 Image Captioning challenge and the State of the Art in its respective category, with a score of 143.7 CIDErD in the offline test split, 140.8 CIDErD in the online evaluation server and 72.9 AllCIDEr on the nocaps validation set. Additionally, we introduce an End to End training algorithm up to 2.8 times faster than established alternatives.

**Index Terms**—Captioning, COCO, Sequence, Expansion

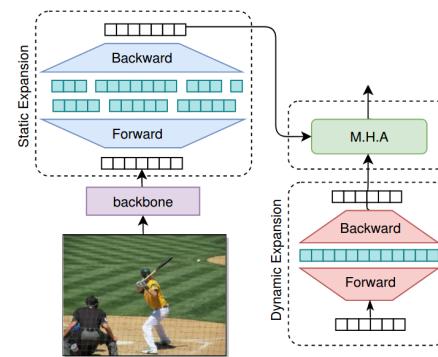
## I. INTRODUCTION

Image Captioning consists of the problem of describing images without human intervention. It is a challenging multimodal task that requires both language comprehension and visual understanding. Early approaches relied on statistical and graph-based methods [1], [2], but since the advent of Neural Networks most Image Captioning systems adopted an encoder and decoder structure [3]–[5]. The first component is responsible for extracting visual features from the image, whereas the latter serves the purpose of generating the description. Early works [3]–[5] relied on Convolutional Neural Network (CNN) backbones [6] combined with Recurrent Neural Networks (RNNs) [7], [8] to further refine the visual inputs and for text generation. In contrast, modern Image Captioning systems adopt Attention-based [9], [10] architectures for the sequence modelling part and, in recent works [11]–[13], also during the image feature extraction. Currently, fully attentive models are the standard de facto architecture in many NLP and Vision research fields and their ubiquity led to many refinements and improvements of the formulation across multiple fields [11], [12], [14]–[20]. However, one of the purposes of the development Attention mechanism [9], [21] was to spread the input sequence content along the whole collection of encoder’s hidden vectors instead of one single state, overcoming a significant performance bottleneck in RNNs. To do so, as the name suggests, the Attention mechanism enhances the values of a few elements and inhibits the others by means of the

Fig. 1: The expansion mechanism distributes the input data into another one featuring a different sequence length during the forward phase and performs the reverse operation in the backward pass. In this way, the network is enabled to process the sequence unconstrained by the number of elements.

Softmax function. Recently, many studies [22]–[27] deepened the understanding of attention approach and suggested that there is little difference between the first and alternative solutions such as Gaussian distributions [25], MLPs [27], Fourier Transform [26] and suggested that the effectiveness of these methods depends mainly on their capability to form high-quality compositions out of the input. Motivated by these observations, our work investigates the possibility that the fixed number of elements provided by the input (the sequence length) represents a performance bottleneck for stateless architectures and limits their potential to form higher-quality compositions, in the particular field of Image Captioning. To this end, we propose the Expansion mechanism, a method that distributes and processes the sequence content using an increased or arbitrary number of elements and retrieves the original length back in the complementary backward operation. We then introduce ExpansionNet v2 (depicted in Fig. 1), which to our knowledge is the first model that learns to exploit arbitrary sequence lengths in Image Captioning and achieves very competitive results without relying on the Attention’s characteristic function.

The overall contributions of this work are the following:



# Khai thác nhiều độ dài chuỗi trong quá trình đào tạo đầu cuối nhanh chóng để thêm chú thích cho hình ảnh

Phòng FIM Jia    
Cheng Hu FIM  
Đại học Modena và Reggio Emilia Modena, Ý  
jiacheng.hu@unimore.it

Roberto Cavicchioli    
Phòng DCE  
Đại học Modena và Reggio Emilia Reggio Emilia, Ý  
roberto.cavicchioli@unimore.it

Phòng FIM Alessandro Capotondi    
Đại học Modena và Reggio Emilia Modena, Ý  
alessandro.capotondi@unimore.it

Tóm tắt—Chúng tôi giới thiệu một phương pháp gọi là Expansion mechanism xử lý dữ liệu đầu vào không bị giới hạn bởi số lượng phần tử trong chuỗi. Bằng cách đó, mô hình có thể học hiệu quả hơn so với các phương pháp tiếp cận dựa trên sự chú ý truyền thống. Để hỗ trợ cho tuyên bố này, chúng tôi thiết kế một kiến trúc mới ExpansionNet v2 đã đạt được kết quả cao trong thử thách Chú thích hình ảnh MS COCO 2014 và State of the Art trong danh mục tương ứng, với số điểm là 143,7 CIDErD trong phần chia tách thử nghiệm ngoại tuyến, 140,8 CIDErD trong máy chủ đánh giá trực tuyến và 72,9 AllCIDEr trên bộ xác thực nocaps. Ngoài ra, chúng tôi giới thiệu một thuật toán đào tạo End to End nhanh hơn tới 2,8 lần so với các giải pháp thay thế đã thiết lập.

Thuật ngữ chỉ mục—Chú thích, COCO, Trình tự, Mở rộng

## I. GIỚI THIỆU

Chú thích hình ảnh bao gồm vấn đề mô tả hình ảnh mà không có sự can thiệp của con người. Đây là một nhiệm vụ đa phương thức đầy thách thức đòi hỏi cả khả năng hiểu ngôn ngữ và hiểu thị giác. Các phương pháp tiếp cận ban đầu dựa trên các phương pháp thống kê và dựa trên đồ thị [1], [2], nhưng kể từ khi Mạng nơ-ron ra đời, hầu hết các hệ thống Chú thích hình ảnh đều áp dụng cấu trúc mã hóa và giải mã [3]–[5]. Thành phần đầu tiên chịu trách nhiệm trích xuất các đặc điểm trực quan từ hình ảnh, trong khi thành phần sau phục vụ mục đích tạo ra mô tả. Các công trình ban đầu [3]–[5] dựa trên xương sống Mạng nơ-ron tích chập (CNN) [6] kết hợp với Mạng nơ-ron hồi quy (RNN) [7], [8] để tinh chỉnh thêm các đầu vào trực quan và để tạo văn bản. Ngược lại, các hệ thống Chú thích hình ảnh hiện đại áp dụng kiến trúc dựa trên Sự chú ý [9], [10] cho phần mô hình hóa chuỗi và, trong các công trình gần đây [11]–[13], cũng trong quá trình trích xuất đặc điểm hình ảnh. Hiện nay, các mô hình hoàn toàn chú ý là kiến trúc de facto tiêu chuẩn trong nhiều lĩnh vực nghiên cứu NLP và Vision và tính phổ biến của chúng đã dẫn đến nhiều cải tiến và tinh chỉnh công thức trên nhiều lĩnh vực [11], [12], [14]–[20]. Tuy nhiên, một trong những mục đích của cơ chế Attention phát triển [9], [21] là phân tán nội dung chuỗi đầu vào đọc theo toàn bộ tập hợp các vectơ ẩn của bộ mã hóa thay vì một trạng thái duy nhất, khắc phục tình trạng tắc nghẽn hiệu suất đáng kể trong RNN. Để làm như vậy, như tên gọi của nó, cơ chế Attention tăng cường giá trị của một số phần tử và ức chế các phần tử khác bằng cách

Công trình này đã nhận được tài trợ từ Chương trình Horizon 2020 của Liên minh Châu Âu chương trình dAIEDGE (GA số 101120726).

Hình 1: Cơ chế mở rộng phân phối dữ liệu đầu vào vào một cơ chế khác có độ dài chuỗi khác nhau trong pha tiền và thực hiện thao tác ngược trong pha lùi. Theo cách này, mạng có thể xử lý chuỗi mà không bị giới hạn bởi số lượng phần tử.

Hàm Softmax. Gần đây, nhiều nghiên cứu [22]–[27] đã đào sâu hiểu biết về phương pháp chú ý và gợi ý rằng có rất ít sự khác biệt giữa các giải pháp đầu tiên và các giải pháp thay thế như phân phối Gaussian [25], MLP [27], Biến đổi Fourier [26] và gợi ý rằng hiệu quả của các phương pháp này phụ thuộc chủ yếu vào khả năng hình thành các thành phần chất lượng cao từ đầu vào. Được thúc đẩy bởi những quan sát này, công trình của chúng tôi điều tra khả năng số lượng phần tử cố định do đầu vào cung cấp (độ dài chuỗi) biểu thị một nút thất hiệu suất cho các kiến trúc không trạng thái và hạn chế tiềm năng của chúng trong việc hình thành các thành phần chất lượng cao hơn, trong lĩnh vực cụ thể là Chú thích hình ảnh. Để đạt được mục đích này, chúng tôi đề xuất Cơ chế mở rộng, một phương pháp phân phối và xử lý nội dung chuỗi bằng cách sử dụng số lượng phần tử tăng lên hoặc tùy ý và truy xuất lại độ dài ban đầu trong phép toán ngược bổ sung. Sau đó, chúng tôi giới thiệu ExpansionNet v2 (được mô tả trong Hình 1), theo chúng tôi biết đây là mô hình đầu tiên học cách khai thác độ dài chuỗi tùy ý trong Chú thích hình ảnh và đạt được kết quả xắt cạnh tranh mà không cần dựa vào hàm đặc trưng của Attention.

Những đóng góp chung của công trình này như sau:

(i) we introduce a new method called Expansion Mechanism that distributes the input content over an arbitrary or increased number of elements during the forward step, and retrieves the original length back in the complementary backward operation. To support both bidirectional and auto-regressive processing, we introduce two methods, called Static Expansion and Dynamic Expansion. The efficiency aspect is addressed in their design and as a result, the computational impact is negligible for small configurations; (ii) with the aforementioned methods, we design a novel architecture called ExpansionNet v2 that achieves strong results on the MS-COCO 2014 outperforming similar models trained on the same dataset; (iii) given the positive results of our architecture, we find out that traditional architectures in Image Captioning are indeed penalized by the fixed number of elements provided by the input; (iv) in contrast to the general trend, our achieves strong results despite the removal of the Attention in most components. Finally, we also propose a fast End-to-End training strategy that lowers significantly the training cost of our model compared to popular approaches.

## II. RELATED WORKS

Image Captioning models benefited greatly from Deep Learning methods. From hand-crafted sentences combined with object detection [28], [29], modern systems consist of a neural encoder that extracts meaningful visual representations from the image and a decoder responsible for the description generation. In the early formulations, the decoder consisted of RNNs [7], [8], whereas the encoder consisted of a convolutional backbone [3], [4] that represented the entire image with a single feature vector. It was later replaced by an object detector [5] that extracted a collection of salient regions of the image. This enabled the adoption of sequence modelling architectures in both encoding and decoding [3]–[5], [30] on top of the backbones. Most modern Image Captioning systems are currently based on the Transformer architecture [10] and many works focused on improving its formulation or structure [14], [17], [19], [31]–[34]. For example, the work of [17] introduced geometrical awareness in the Self-Attention formulation. [31] modified the attentive layer with a gate that served the purpose of mitigating the contribution of irrelevant queries. [14] exploited the bilinear pooling to enable a higher order of interactions across the input elements. Other works such as [12], [18], [35] focused on structural changes and exploiting the visual input more effectively. Overall, all these methods follow the main components of the formulas introduced in [9], [10], [21]. Our Expansion mechanism is based on the adoption of embedding vectors. The effectiveness of integrating additional learnable parameters in the sequence was observed first in [33] in Machine Translation. Later in Image Captioning, the concept was also deployed by [36] and [37]. In contrast to these works, our method is the only one that distributes the input into an arbitrary number of hidden vectors.

Another trend consists of pre-training the model with a huge amount of training data and fine-tuning over the Im-

age Captioning task [16], [38]–[40]. In particular, OFA [38] and GIT [16] currently represent the State-of-the-Art Image Captioning systems and outperform non-generative models by a significant margin. However, their model size poses an obstacle to the deployment in memory-limited devices and the training data are tens and hundreds of times bigger than the popular MS-COCO 2014 [41]. For this reason, these works are considered orthogonal to ours which can be instead integrated to potentially achieve better performances. In general, we only consider works that are trained exclusively on MS-COCO 2014, for this reason, the works of [12], [15], [16], [39], [40] are omitted during evaluation since our model does not leverage additional data.

## III. METHOD

### A. Static and Dynamic Expansion

The Expansion mechanism is broken down into several steps. First, it distributes the sequence content into an arbitrary or increased number of elements (Section III-A1) using a “Forward Expansion”, which is described in Section III-A2 and allows the network to process the sequence unconstrained by the fixed input length. Then, it retrieves the original length using the complementary operation “Backward Expansion”, described in Section III-A3. Depending on the operations, we define two implementations of the idea: Static Expansion and Dynamic Expansion. The latter is designed to support both the auto-regressive and bidirectional processing, in contrast to the first, which only supports the bidirectional case.

1) *Expansion coefficient*: In both Static and Dynamic Expansion, the expansion coefficient  $N_E$  defines a collection of learnable parameters  $E_Q, E_B \in \mathbb{R}^{N_E \times d_m}$ . However, in the Static Expansion,  $N_E$  defines exactly the size of the expanded sequences regardless of the input length  $L$ . In particular, the expansion queries  $Q_E$  and biases  $B_E$  equal to  $E_Q$  and  $E_B$  respectively. In contrast, in the Dynamic Expansion, the expanded sequence is of size  $N_E \cdot L$ , and the expansion queries  $Q_E^D$  and biases  $B_E$  are calculated with the BroadSum operator, defined in the two cases as:

$$\begin{aligned} Q_E &= (C^\top \mathbb{H}_E)^\top + (E_Q^\top \mathbb{I}_E)^\top \\ B_E &= (C^\top \mathbb{H}_E)^\top + (E_B^\top \mathbb{I}_E)^\top \end{aligned} \quad (1)$$

where  $C \in \mathbb{R}^{L \times d_m}$  denotes a linear projection of the input and  $\mathbb{H}_E \in \mathbb{R}^{L \times (L \cdot N_E)}$  is defined as:

$$\mathbb{H}_E = \begin{bmatrix} 1 & 0 & \dots & 0 \\ 0 & 1 & \dots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \dots & 1 \end{bmatrix}, \quad 1, 0 \in \mathbb{R}^{1 \times N_E}$$

whereas  $\mathbb{I}_E \in \mathbb{R}^{N_E \times (L \cdot N_E)}$  is defined by the column-wise concatenation of  $L$  identity matrices of size  $N_E \times N_E$ :

$$\mathbb{I}_E = [\mathbb{I}_L \quad \mathbb{I}_L \quad \dots \quad \mathbb{I}_L], \quad \mathbb{I}_L \in \mathbb{R}^{N_E \times N_E}$$

An example of the input and output of the BroadSum operation is depicted in the bottom left of Figure 2, where the bias vectors are omitted for simplicity.

(i) chúng tôi giới thiệu một phương pháp mới gọi là Expansion Mechanism phân phối nội dung đầu vào trên một số lượng phần tử tùy ý hoặc tăng lên trong bước tiến và truy xuất lại độ dài ban đầu trong thao tác ngược bổ sung. Để hỗ trợ cả xử lý song hướng và tự hồi quy, chúng tôi giới thiệu hai phương pháp, được gọi là Static Expansion và Dynamic Expansion. Khía cạnh hiệu quả được giải quyết trong thiết kế của chúng và do đó, tác động tính toán là không đáng kể đối với các cấu hình nhỏ; (ii) với các phương pháp đã đề cập ở trên, chúng tôi thiết kế một kiến trúc mới gọi là ExpansionNet v2 đạt được kết quả mạnh mẽ trên MS-COCO 2014 vượt trội hơn các mô hình tương tự được đào tạo trên cùng một tập dữ liệu; (iii) với kết quả tích cực của kiến trúc của chúng tôi, chúng tôi thấy rằng các kiến trúc truyền thống trong Image Captioning thực sự bị phạt do số lượng phần tử cố định do đầu vào cung cấp; (iv) trái ngược với xu hướng chung, chúng tôi đạt được kết quả mạnh mẽ mặc dù loại bỏ Attention trong hầu hết các thành phần. Cuối cùng, chúng tôi cũng đề xuất một chiến lược đào tạo End-to-End nhanh chóng giúp giảm đáng kể chi phí đào tạo mô hình của chúng tôi so với các phương pháp phổ biến.

## II. CÁC TÁC PHẨM LIÊN QUAN

Các mô hình chú thích hình ảnh được hưởng lợi rất nhiều từ các phương pháp học sâu. Từ các câu được tạo thủ công kết hợp với phát hiện đối tượng [28], [29], các hệ thống hiện đại bao gồm một bộ mã hóa nơ-ron trích xuất các biểu diễn trực quan có ý nghĩa từ hình ảnh và một bộ giải mã chịu trách nhiệm tạo ra mô tả. Trong các công thức ban đầu, bộ giải mã bao gồm RNN [7], [8], trong khi bộ mã hóa bao gồm một xung sóng tích chất [3], [4] biểu diễn toàn bộ hình ảnh bằng một vectơ đặc trưng duy nhất. Sau đó, nó được thay thế bằng một bộ phát hiện đối tượng [5] trích xuất một tập hợp các vùng nổi bật của hình ảnh. Điều này cho phép áp dụng các kiến trúc mô hình hóa chuỗi trong cả mã hóa và giải mã [3]–[5], [30] trên các xung sóng. Hầu hết các hệ thống chú thích hình ảnh hiện đại hiện nay đều dựa trên kiến trúc Transformer [10] và nhiều công trình tập trung vào việc cải thiện công thức hoặc cấu trúc của nó [14], [17], [19], [31]–[34]. Ví dụ, công trình của [17] đã giới thiệu nhận thức hình học trong công thức Tự chủ ý. [31] đã sửa đổi lớp chú ý bằng một công thức mục đích giám thiểu sự đóng góp của các truy vấn không liên quan. [14] đã khai thác nhón song tuyến tính để cho phép thử tự tương tác cao hơn trên các phần tử đầu vào. Các công trình khác như [12], [18], [35] tập trung vào các thay đổi về cấu trúc và khai thác đầu vào trực quan hiệu quả hơn. Nhìn chung, tất cả các phương pháp này đều tuân theo các thành phần chính của các công thức được giới thiệu trong [9], [10], [21]. Cơ chế mở rộng của chúng tôi dựa trên việc áp dụng các vectơ nhúng. Hiệu quả của việc tích hợp các tham số có thể học được bổ sung trong chuỗi đã được quan sát lần đầu tiên trong [33] trong Dịch máy. Sau đó trong Chú thích hình ảnh, khái niệm này cũng được triển khai bởi [36] và [37]. Trái ngược với các công trình này, phương pháp của chúng tôi là phương pháp duy nhất phân phối đầu vào thành một số vectơ ẩn tùy ý.

Nhiệm vụ chú thích tuổi [16], [38]–[40]. Đặc biệt, OFA [38] và GIT [16] hiện đang đại diện cho các hệ thống chú thích hình ảnh hiện đại và vượt trội hơn các mô hình không tạo ra với biên độ dài ban đầu trong thao tác ngược bổ sung. Để hỗ trợ cả xử lý song hướng và tự hồi quy, chúng tôi giới thiệu hai phương pháp, được gọi là Static Expansion và Dynamic Expansion. Khía cạnh hiệu quả được giải quyết trong thiết kế của chúng và do đó, tác động tính toán là không đáng kể đối với các cấu hình nhỏ; (ii) với các phương pháp đã đề cập ở trên, chúng tôi thiết kế một kiến trúc mới gọi là ExpansionNet v2 đạt được kết quả mạnh mẽ trên MS-COCO 2014 vượt trội hơn các mô hình tương tự được đào tạo trên cùng một tập dữ liệu; (iii) với kết quả tích cực của kiến trúc của chúng tôi, chúng tôi thấy rằng các kiến trúc truyền thống trong Image Captioning thực sự bị phạt do số lượng phần tử cố định do đầu vào cung cấp; (iv) trái ngược với xu hướng chung, chúng tôi đạt được kết quả mạnh mẽ mặc dù loại bỏ Attention trong hầu hết các thành phần. Cuối cùng, chúng tôi cũng đề xuất một chiến lược đào tạo End-to-End nhanh chóng giúp giảm đáng kể chi phí đào tạo mô hình của chúng tôi so với các phương pháp phổ biến.

## III. PHƯƠNG PHÁP

### A. Mở rộng tĩnh và động Cơ chế mở

rộng được chia thành nhiều bước. Đầu tiên, nó phân phối nội dung chuỗi thành một số phần tử tùy ý hoặc tăng lên (Phần III-A1) bằng cách sử dụng “Mở rộng tĩnh”, được mô tả trong Phần III-A2 và cho phép mạng xử lý chuỗi không bị giới hạn bởi độ dài đầu vào cố định. Sau đó, nó truy xuất độ dài ban đầu bằng cách sử dụng thao tác bổ sung “Mở rộng lùi”, được mô tả trong Phần III-A3. Tùy thuộc vào các thao tác, chúng tôi định nghĩa hai triển khai của ý tưởng: Mở rộng tĩnh và Mở rộng động. Sau này được thiết kế để hỗ trợ cả xử lý tự hồi quy và song hướng, trái ngược với đầu tiên, chỉ hỗ trợ trường hợp song hướng.

1) *Hệ số mở rộng*: Trong cả Static và Dynamic Expansion, hệ số mở rộng NE định nghĩa một tập hợp các tham số có thể học được EQ, EB  $\in \mathbb{R}^{N_E \times d_m}$ . Tuy nhiên, trong Static Expansion, NE định nghĩa chính xác kích thước của các chuỗi mở rộng bất kể độ dài đầu vào L. Cụ thể, các truy vấn mở rộng QE và bias BE tương ứng bằng EQ và EB. Ngược lại, trong Dynamic Expansion, chuỗi mở rộng có kích thước  $NE \cdot L$  và các truy vấn mở rộng QE và bias BE được tính toán bằng toán tử BroadSum, được định nghĩa trong hai trường hợp như sau:

$$\begin{aligned} QE &= (C^\top HE)^\top + (E^\top QIE)^\top \\ BE &= (C^\top HE)^\top + (E^\top (t_{\text{tử}})_E)^\top \end{aligned} \quad (1)$$

trong đó  $C \in \mathbb{R}^{L \times d_m}$  biểu thị phép chiếu tuyến tính của đầu vào và  $HE \in \mathbb{R}^{L \times (L \cdot NE)}$  được định nghĩa là:

$$\text{Anh} \hat{a}y = \begin{bmatrix} 1 & 0 & \dots & 0 \\ 0 & 1 & \dots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \dots & 1 \end{bmatrix}, \quad 1, 0 \in \mathbb{R}^{1 \times DB}$$

trong khi  $IE \in \mathbb{R}^{NE \times (L \cdot NE)}$  được xác định bằng cách nối từng cột của L ma trận dòng nhất có kích thước  $NE \times NE$ :

$$IE = IL \quad IL \quad \dots \quad THE, \quad IL \in \mathbb{R}^{NE \times NE}$$

Một ví dụ về đầu vào và đầu ra của phép toán BroadSum được mô tả ở góc dưới bên trái của Hình 2, trong đó các vectơ độ lệch được bỏ qua để đơn giản hóa.

Một xu hướng khác bao gồm việc đào tạo trước mô hình với một lượng lớn dữ liệu đào tạo và tinh chỉnh trong Im-

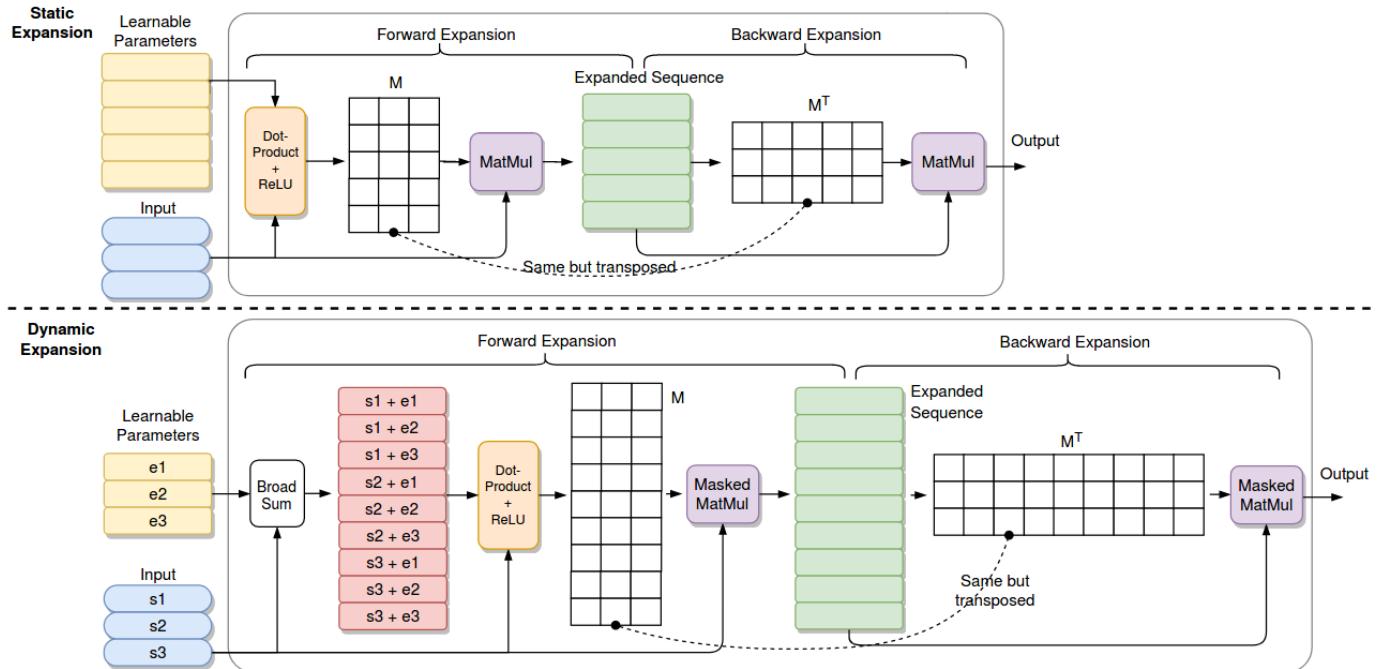


Fig. 2: Static Expansion and Auto-regressive Dynamic Expansion scheme and example. Assuming an input length of  $L = 3$ . In the Static Expansion setting, an expansion coefficient of  $N_E = 5$  leads to an expanded sequence of length 5. In contrast, in the Dynamic Expansion, an expansion coefficient of  $N_E = 3$  generates an expanded sequence of  $L \cdot N_E = 9$ . For the sake of simplicity, the double operation stream, the expansion biases and the gated result combination are omitted in the illustration. The difference between the Auto-regressive Dynamic Expansion and the bidirectional one lies in the Masked Matrix Multiplication.

2) *Forward Expansion*: The forward expansion generates the expanded sequences and involves three linear projections of the input, denoted as  $K, V_1, V_2 \in \mathbb{R}^{L \times d_m}$ . First of all, the “Length Transformation Matrix”, denoted as  $M$ , is computed as the dot-product similarity between  $K$  and the expansion queries  $Q_E$ :

$$M = \frac{Q_E K^\top}{\sqrt{d_m}}. \quad (2)$$

The result is fed into the following operations:

$$R_i^{bw} = \Psi(ReLU((-1)^i M), \epsilon) \quad i \in \{1, 2\} \quad (3)$$

where  $\Psi : (X, \epsilon) \rightarrow Y, X, Y \in \mathbb{R}^{N_1 \times N_2}, \epsilon \in \mathbb{R}^+ / \{0\}$  is the row-wise normalization function defined as:

$$\Psi(X, \epsilon)_{ij} = \frac{x_{ij}}{\sum_{z=1}^{N_2} x_{iz} + \epsilon} \quad (4)$$

the coefficient  $\epsilon$  ensures the feasibility of the operation. Then, the expanded sequences are calculated as follows:

$$F_i^{fw} = R_i^{bw} V_i + B_E \quad i \in \{1, 2\} \quad (5)$$

3) *Backward expansion*: In the backward step, the original sequence length is retrieved by transposing the length transformation matrix in Equation 2 and applying the same operations of Equation 3:

$$R_i^{bw} = \Psi(ReLU((-1)^i M^\top), \epsilon) \quad i \in \{1, 2\} \quad (6)$$

This time, the matrices  $R_i^{bw}$  are multiplied with the expanded sequences of Equation 5:

$$B_i^{bw} = R_i^{bw} F_i^{fw} \quad i \in \{1, 2\} \quad (7)$$

Finally, the final results  $B_1^{bw}$  and  $B_2^{bw}$  are combined by means of a sigmoid gate:

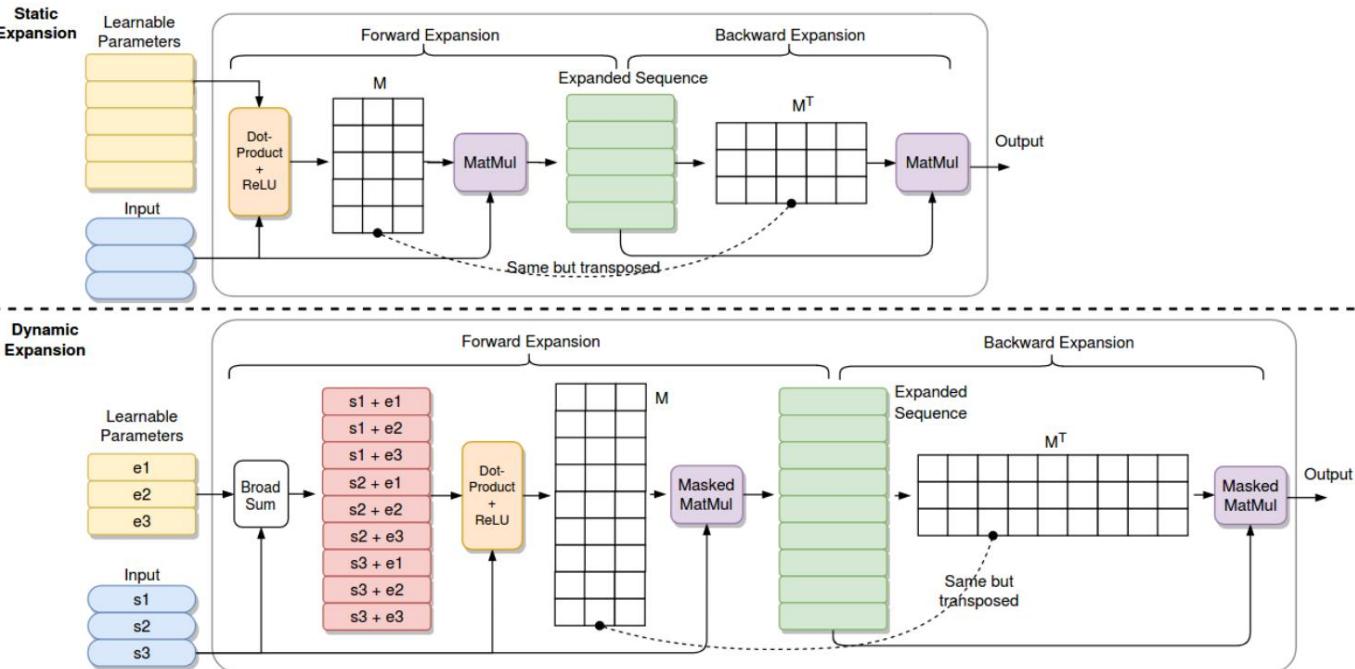
$$out = \sigma(S) \odot B_1^{bw} + (1 - \sigma(S)) \odot B_2^{bw}. \quad (8)$$

where  $S \in \mathbb{R}^L$  is a linear projection of the input.

The backward operation completes the operations performed in the Static and Dynamic Expansion. It can be noted that all operations in the forward (3)(5) and backward expansion (7)(6) are duplicated in two operations streams for  $i = 1$  and  $i = 2$ , differing mainly in the sign of the computation of the Length Transformation Matrix in (2). The decision was made to mitigate the remote possibility of the matrix being populated only by zeros. This does not affect the results compared to the single path but slightly increases the computational cost.

In the case of Dynamic Expansion, masking is applied when calculating the results in (5) and (7) to preserve the auto-regressive property. The operation principle of Static and Dynamic Expansions are illustrated in Fig. 2, which, for simplicity, depicts only a single operation stream and omits biases and the output sigmoid gates.

4) *Block Static Expansion*: To increase the effectiveness of the Static Expansion, we perform the Forward and Backward operations on a collection of target lengths instead of one. We



Hình 2: Sơ đồ và ví dụ về Mở rộng tĩnh và Mở rộng động tự hồi quy. Giả sử chiều dài đầu vào là  $L = 3$ . Trong cài đặt Static Expansion, hệ số mở rộng  $N_E = 5$  dẫn đến một chuỗi mở rộng có độ dài 5. Ngược lại, trong Dynamic Expansion, hệ số mở rộng  $N_E = 3$  tạo ra một chuỗi mở rộng  $L \cdot N_E = 9$ . Để đơn giản, luồng hoạt động kép, độ lệch mở rộng và kết hợp kết quả có công được bỏ qua trong hình minh họa. Sự khác biệt giữa Auto-regressive Dynamic Expansion và bidirectional Expansion nằm ở Masked Matrix Multiplication.

2) Mở rộng về phía trước: Mở rộng về phía trước tạo ra các chuỗi mở rộng và bao gồm ba phép chiếu tuyến tính của đầu vào, được ký hiệu là  $K, V_1, V_2 \in \mathbb{R}^{L \times d_m}$ . Trước hết, “Ma trận chuyển đổi độ dài”, được ký hiệu là  $M$ , được tính là độ tương tự tích vô hướng giữa  $K$  và các truy vấn mở rộng QE:

$$M = \frac{QEK}{\sqrt{d_m}}. \quad (2)$$

Kết quả được đưa vào các hoạt động sau:

$$R_{tối}^{tối} = \Psi(ReLU((-1)iM), \epsilon) \quad i \in \{1, 2\} \quad (3)$$

đó  $\Psi : (X, \epsilon) \rightarrow Y$  hàm chuẩn,  $X, Y \in \mathbb{R}^{N_1 \times N_2}$  trong  $\epsilon \in \mathbb{R}^+ / \{0\}$  là hóa theo hàng được định nghĩa như sau:

$$\Psi(X, \epsilon)_{ij} = \frac{x_{ij}}{\sum_{z=1}^{N_2} x_{iz} + \epsilon} \quad (4)$$

hệ số  $\epsilon$  đảm bảo tính khả thi của phép toán. Sau đó, các chuỗi mở rộng được tính như sau:

$$F_{tối}^{fw} = R_{tối}^{tối} V_i + B_E \quad i \in \{1, 2\} \quad (5)$$

3) Mở rộng ngược: Trong bước ngược, độ dài chuỗi ban đầu được lấy lại bằng cách chuyển vị ma trận chuyển đổi độ dài trong Phương trình 2 và áp dụng các phép toán tương tự của Phương trình 3:

$$R_{tối}^{tối} = \Psi(ReLU((-1)iM^\top), \epsilon) \quad i \in \{1, 2\} \quad (6)$$

Lần này, các ma trận  $R_{bw}$  được nhân với các chuỗi mở rộng của Phương trình 5:

$$B_{tối}^{bw} = R_{tối}^{tối} \quad \text{chi tiết} \quad i \in \{1, 2\} \quad (7)$$

Cuối cùng, kết quả cuối cùng  $B_{bw}$  và  $B_{bw}^*$  được kết hợp bằng công thức:

$$ra = \sigma(S) \odot B_1^{bw} + (1 - \sigma(S)) \odot B_2^{bw}. \quad (8)$$

trong đó  $S \in \mathbb{R}^L$  là phép chiếu tuyến tính của đầu vào. Phép toán ngược hoàn tất các phép toán được thực hiện trong Phép toán mở rộng tĩnh và động. Có thể lưu ý rằng tất cả các phép toán trong phép toán mở rộng tiên (3)(5) và phép toán mở rộng lùi (7)(6) đều được nhân đổi trong hai luồng phép toán đổi với  $i = 1$  và  $i = 2$ , chủ yếu khác nhau về dấu của phép toán. Ma trận biến đổi độ dài trong (2). Quyết định được đưa ra là để giảm thiểu khả năng từ xa của ma trận chỉ được di chuyển số không. Điều này không ảnh hưởng đến kết quả so với đường dẫn đơn nhưng làm tăng nhẹ chi phí tính toán.

Trong trường hợp của Dynamic Expansion, việc che dấu được áp dụng khi tính toán kết quả trong (5) và (7) để bảo toàn tính chất tự hồi quy. Nguyên lý hoạt động của Static Expansion và Dynamic Expansion được minh họa trong Hình 2, vì mục đích đơn giản, hình này chỉ mô tả một luồng hoạt động duy nhất và bỏ qua các độ lệch và công sigmoid đầu ra.

4) Mở rộng tĩnh khôi: Để tăng hiệu quả của Mở rộng tĩnh, chúng tôi thực hiện các hoạt động Tiền và Lùi trên một tập hợp các độ dài mục tiêu thay vì một. Chúng tôi

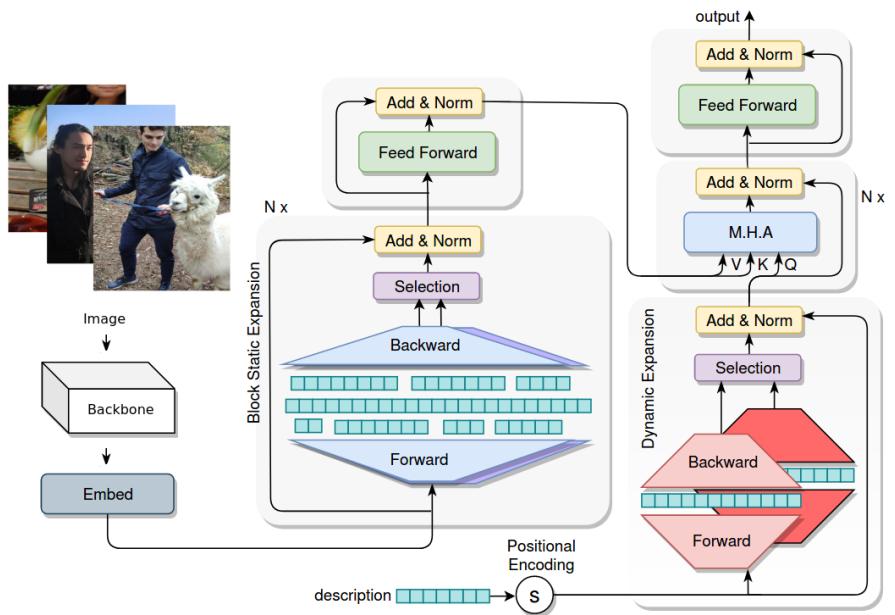


Fig. 3: ExpansionNet v2 architecture.

call the operation Block Static Expansion. From a formulation perspective, all operations are repeated over a group of expansion coefficients  $G = \{N_E^1, N_E^2, \dots, N_E^{N_G}\}$  and can be implemented in a way such that both forward and backward steps are performed over all targets at the same time. All expansion group queries and biases can be combined into a single one:

$$\begin{aligned} E_Q^G &= \{(E_Q^1)^\top, (E_Q^2)^\top, \dots, (E_Q^{N_G})^\top\}^\top \\ E_B^G &= \{(E_B^1)^\top, (E_B^2)^\top, \dots, (E_B^{N_G})^\top\}^\top \end{aligned} \quad (9)$$

and the computational efficiency of the previous formulation can be preserved. During the backward stage, the length transformation matrix is scaled by the inverse number of elements in the group  $G$ .

### B. Architecture

Our model consists of the standard encoder-decoder structure implemented on top of the Swin-Transformer, which details are provided in [13]. The image  $A$  is first fed into the backbone:

$$X_0 = \text{Swin-Transf}(A) \quad (10)$$

and generates the initial set of processed visual features  $X_0 = \{x_1^0, x_2^0, \dots, x_N^0\}$ ,  $x_i^0 \in \mathbb{R}^{d_m}$ . The result is fed into the encoder, which is made of  $N_{enc}$  Static Expansion  $\rightarrow$  FeedForward blocks. Here skip connection and pre-layer normalization [42] are adopted, and the following formulas describe each encoder layer for  $n \in \{1, \dots, N_{enc}\}$ :

$$\begin{aligned} E_n &= X_{n-1} + \text{StaticExp}_n(\text{Norm}_n^{SE}(X_{n-1})) \\ X_n &= E_n + FF_n(\text{Norm}_n^{FF}(B_n)) \end{aligned} \quad (11)$$

Similarly, given a generic input sequence  $Y_0 = \{y_1^0, y_2^0, \dots, y_M^0\}$ ,  $y_i^0 \in \mathbb{R}^{d_m}$  (at training stage so we

can omit the time axis), the decoder is made of  $N_{dec}$  Dynamic Expansion  $\rightarrow$  Cross-Attention  $\rightarrow$  FeedForward blocks, where skip connection and normalization is applied on each component. Each decoder layer is described by the following equations:

$$\begin{aligned} B_n &= Y_{n-1} + \text{DynamicExp}_n(\text{Norm}_n^{DE}(Y_{n-1})) \\ W_n &= B_n + \text{Attention}_n(\text{Norm}_n^{CA}(B_n), X_{N_{enc}}) \\ Y_n &= W_n + FF_n(\text{Norm}_n^{FF}(W_n)) \end{aligned} \quad (12)$$

All layers are summed through a linear projection and the final output is fed to the classification layer. Fig. 3 depicts the main structure.

### C. Training objectives

The model is first pre-trained using the Cross-Entropy loss  $L_{XE}$ :

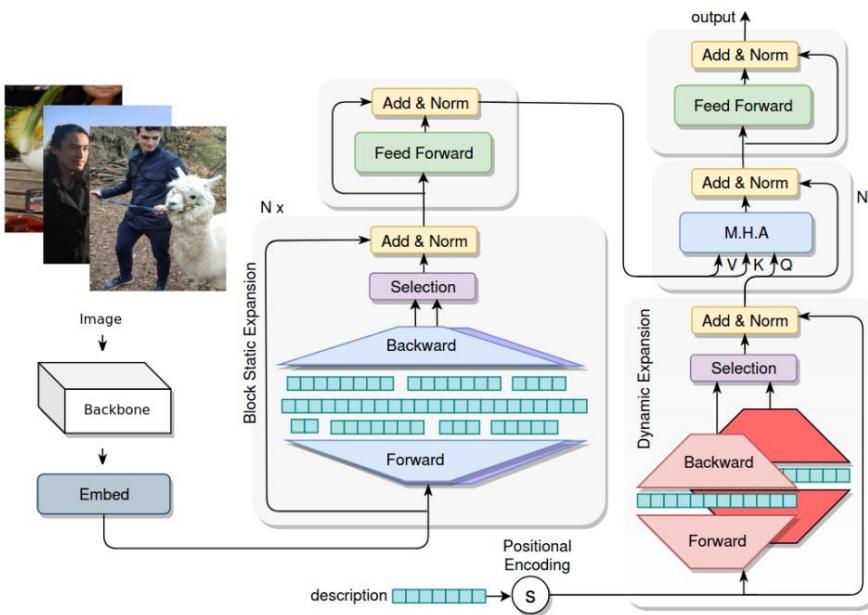
$$L_{XE}(\theta) = - \sum_t^T \log(p_\theta(y_t^* | y_{1:t-1}^*, I)) \quad (13)$$

where  $p_\theta(y_t^* | y_{1:t-1}^*, I)$  is the probability assigned by the model parameters  $\theta$  to the target  $y_t^*$  given the image  $I$  and the previous words  $y_{1:t-1}^*$ . Additionally, the CIDEr-D score is optimized using the SCST [43] which minimizes the negative expected reward  $L_R(\theta) = -\mathbb{E}_{y_{1:T} \sim p_\theta}[r(y_{1:T})]$ , which gradient can be approximated as follows:

$$\nabla_\theta L_R(\theta) \approx -(r(y_{1:T}^s) - b) \nabla_\theta \log p_\theta(y_{1:T}^s) \quad (14)$$

$b$  is the baseline computed according to [44] and  $r(y_{1:T}^s)$  is the CIDEr-D reward assigned to the sampled sequence  $y_{1:T}^s$ .

Although we optimize the model on two loss functions, for each one of them, the training stage is efficiently split into two additional steps to allow a broader number of computational resources to reproduce this work.



Hình 3: Kiến trúc ExpansionNet v2.

gọi hoạt động Block Static Expansion. Theo quan điểm công thức, tất cả các hoạt động được lặp lại trên một nhóm các hệ số mở rộng  $G = \{N1 \text{ được triển khai theo cách } v, N2 \text{ và }, \dots, N_{QF}\}$  và có thể sao cho cả các bước tiến và lùi đều được thực hiện trên tất cả các mục tiêu cùng một lúc. Tất cả các truy vấn và độ lệch của nhóm mở rộng có thể được kết hợp thành một truy vấn duy nhất:

$$\begin{aligned} V_A^G &= \{V_A^1, V_A^2, \dots, V_A^{N_G}\} \\ V_B^G &= \{V_B^1, V_B^2, \dots, V_B^{N_G}\} \end{aligned} \quad (9)$$

((E và hiệu quả tính toán của công thức trước đó có thể được bảo toàn. Trong giai đoạn ngược lại, ma trận biến đổi độ dài được chia tỷ lệ theo số nghịch đảo các phần tử trong nhóm G.

### B. Kiến trúc

Mô hình của chúng tôi bao gồm cấu trúc mã hóa-giải mã tiêu chuẩn được triển khai trên Swin-Transformer, thông tin chi tiết được cung cấp trong [13]. Hình ảnh A đầu tiên được đưa vào xướng sống:

$$X_0 = \text{Chuyển đổi Swin-T}(A) \quad (10)$$

và tạo ra tập hợp ban đầu các đặc điểm hình ảnh đã xử lý  $\{x_1^0, x_2^0, \dots, x_N^0\}$ . Kết quả được đưa vào  $X_0 = \{x_1^0, x_2^0, \dots, x_N^0\}$  vào bộ mã hóa, được tạo thành từ các khối  $N_{enc}$  Static Expansion FeedForward. Ở đây, kết nối bù qua và chuẩn hóa lớp trước [42] được áp dụng và các công thức sau đây mô tả từng lớp bộ mã hóa cho  $n \in \{1, \dots, N_{enc}\}$ :

$$\begin{aligned} E_n &= X_{n-1} + \text{StaticExp}_n(\text{Norm}_n^{SE}(X_{n-1})) \\ X_n &= E_n + FF_n(\text{Norm}_n^{FF}(B_n)) \end{aligned} \quad (11)$$

Tương tự như vậy, với một chuỗi đầu vào chung  $Y_0 = \{y_1^0, y_2^0, \dots, y_M^0\}$  (ở

có thể bỏ qua trực thời gian), bộ giải mã được tạo thành từ các khối  $N_{dec}$  Dynamic Expansion  $\rightarrow$  Cross-Attention  $\rightarrow$  FeedForward, trong đó kết nối bù qua và chuẩn hóa được áp dụng cho từng thành phần. Mỗi lớp bộ giải mã được mô tả bằng các phương trình sau:

$$B_n = Y_{n-1} + \text{DynamicExp}_n(\text{Norm}_n^{DE}) \quad (Trong 1)$$

$$W_n = B_n + \text{Attention}_n(\text{Norm}_n^{CA}(B_n), X_{N_{enc}}) \quad (12)$$

$$In = W_n + FF_n(\text{Norm}_n^{FF}) \quad (Vàng) \quad (12)$$

Tất cả các lớp được tổng hợp thông qua phép chiếu tuyến tính và đầu ra cuối cùng được đưa vào lớp phân loại. Hình 3 mô tả cấu trúc chính.

### C. Mục tiêu đào tạo Đầu

tiên, mô hình được đào tạo trước bằng cách sử dụng Cross-Entropy loss  $L_{XE}$ :

$$L_{XE}(\theta) = \frac{1}{T} \sum_t \log(p_\theta(y_t | y_{1:t-1}, I)) \quad (13)$$

trong đó  $p_\theta(y_t | y_{1:t-1}, I)$  là xác suất được chỉ định bởi các tham số mô hình  $\theta$  cho mục tiêu  $y_t$  với hình ảnh  $I$  và các từ  $y$  trước đó được tối ưu hóa  $1:t-1$ . Ngoài ra, điểm số CIDEr-D là bằng SCST [43] giúp giảm thiểu phần thưởng dự kiến âm  $LR(\theta) = \mathbb{E}_{y_{1:T} \sim p_\theta}[r(y_{1:T})]$ , có thể xấp xỉ gradient như sau:

$$\nabla_\theta LR(\theta) \approx (r(y_{1:T}) - b) \nabla_\theta \log p_\theta(y_{1:T}) \quad (14)$$

$b$  là đường cơ sở được tính toán theo [44] và  $r(y_t)$  là phần  $t$  của phần thưởng CIDEr-D được gán cho chuỗi lấy mẫu  $y$ .

Mặc dù chúng tôi tối ưu hóa mô hình trên hai hàm mất mát, đối với mỗi hàm, giai đoạn đào tạo được chia thành hai bước bổ sung hiệu quả để cho phép nhiều tài nguyên tính toán hơn có thể tái tạo công việc này.

TABLE I: Ablation study in the first stage of Cross-Entropy training using beam size 3 over the Karpathy validation split. B=BLEU. M=METEOR. R=ROUGE. C=CIDEr-D. S=SPICE.

Encoder	Decoder	B1	B2	B3	B4	M	R	C	S
Baseline	Baseline	75.3	59.2	45.4	34.6	28.4	57.0	115.8	21.6
Stc. Exp. G={16}	Baseline	76.4	60.6	46.6	35.5	28.6	57.2	117.8	21.9
Stc. Exp. G={32}	Baseline	75.9	59.9	46.1	35.2	28.9	57.1	117.9	22.3
Stc. Exp. G={64}	Baseline	76.3	60.4	46.4	35.5	28.8	57.1	117.7	22.0
Baseline	Dyn. Exp. $N_E=4$	77.2	61.4	47.4	36.2	28.9	57.7	119.7	22.3
Baseline	Dyn. Exp. $N_E=8$	76.9	61.5	47.9	37.1	29.1	57.8	120.8	22.3
Baseline	Dyn. Exp. $N_E=16$	76.7	61.4	47.8	36.8	29.0	57.7	121.2	22.2
Stc. Exp. G={64}	Dyn. Exp. $N_E=16$	77.4	61.9	48.2	37.3	29.2	58.0	122.2	22.3
Stc. Exp. G={128, 128, 128, 128, 128}	Dyn. Exp. $N_E=16$	77.8	62.3	48.3	37.2	29.3	58.3	122.8	22.5
Stc. Exp. G={256, 256, 256, 256, 256}	Dyn. Exp. $N_E=16$	77.4	62.0	48.2	37.2	29.2	58.0	122.5	22.2
Stc. Exp. G={512, 512, 512, 512, 512}	Dyn. Exp. $N_E=16$	77.3	61.7	47.9	37.0	29.3	58.0	122.7	22.4
Stc. Exp. G={32, 64, 128, 256, 512}	Dyn. Exp. $N_E=16$	77.6	62.0	48.2	37.2	29.4	58.1	123.5	22.5

## IV. RESULTS

### A. Experimental Setup

1) *Dataset*: The training dataset consists of the popular MS-COCO benchmark [41] split according to [45], resulting in 113287 image-description pairs for the training, 5000 in the validation set, and in the 5000 test set. Each reference caption is pre-processed by a simple pipeline consisting of lowering casing, removing punctuation, and filtering out words that do not occur at least 5 times (vocabulary of size 10000). Additionally, the final model is evaluated over the Novel Object Captioning at Scale (nocaps) dataset validation set [46], which consists of three classes of images called in-domain, near-domain, and out-domain, according to the familiarity of the classes with respect to those contained in the training set. This dataset is subject to the same pre-processing of MS-COCO and serves the purpose of further challenging the model in unfavourable conditions.

2) *Model details*: Two models are implemented for the experimental setup. The baseline, which is the Base Transformer and our main model, referred to as “ExpansionNet v2”, is implemented with the following configurations  $d_m=512$ ,  $d_{ff}=2048$ ,  $N_{enc}=N_{dec}=3$ . In the latter, the Dynamic expansion coefficient is set to 16, and the Static expansion coefficients consist of  $G=\{32, 64, 128, 256, 512\}$  (more details in Section IV-B). Each one relies on top of the same backbone, the Swin-Transformer in the Large configuration [13] pre-trained on ImageNet [47]. All images are subject to a minimal pre-processing: first, they are resized into a  $3\times 384\times 384$  tensor, then RGB values are converted into a  $[0, 1]$  range and further normalized using  $mean=(0.485, 0.456, 0.406)$  and  $std=(0.229, 0.224, 0.225)$ . The source code of the experiments is available<sup>1</sup>.

3) *Training algorithm*: It can be observed that the Swin-Transformer backbone is the most computationally expensive part of the system. For this reason, inspired by [48], to enable the End to End training step to a broader number of computational architectures, our training is divided into four steps in particular, each phase (in both the cross-entropy training and the reinforcement stage) consists of initial training

in which the backbone’s weights are frozen and a fine-tuning step during which gradients flow throughout the whole system:

*Step A) Cross-Entropy – Freezed backbone*. The model is trained using batch size 48, an initial learning rate of  $2e-4$ , a warmup of 10000, and is annealed by 0.8 every 2 epochs for 8 epochs;

*Step B) Cross Entropy – End to End*. The whole system is trained for 2 additional epochs, using batch size 48 and an initial learning rate of  $3e-5$  annealed by 0.55 every epoch;

*Step C) CIDEr-D optimization – Freezed backbone*. Reinforcement phase adopts a batch size of 48, an initial learning rate of  $1e-4$ , no warmup, annealed by 0.8 every epoch for 9 epochs;

*Step D) CIDEr-D optimization – End to End*. The whole system is fine-tuned for a few more iterations up to an additional epoch using a batch size of 20 and fixed learning rate  $2e-6$ . This step is optional since it only slightly contributes to the final performances and can be skipped if no improvements are observed. All CIDEr-D optimization steps are implemented according to the Standard configuration<sup>2</sup>.

Despite its apparent complexity, it is much more computationally friendly than the standard method consisting of a small batch size of 10 for 30 epochs for both optimization steps. As a matter of fact, only a much smaller number of training epochs are dedicated to fine-tuning the whole system. Thus, the time required for the calculation of the backbone’s gradient is often avoided and the time required for forward operations can be drastically reduced as well. In particular, in our implementation, during steps 1 and 3 the backbone’s forward pass is performed only once for each image in the data set. Therefore, its cost is replaced by a memory read and copy. All steps are trained using the RAdam optimizer [53] ( $\beta_1 = 0.9$ ,  $\beta_2 = 0.98$ ).

### B. Ablation Study

To study the effectiveness of our method we replace the encoder and decoder in the baseline with our methods and evaluate several settings of expansion coefficients. It can be observed from Table I that the impact of the static expansion

<sup>1</sup>Code available at: [https://github.com/jchenghu/ExpansionNet\\_v2](https://github.com/jchenghu/ExpansionNet_v2)

<sup>2</sup>SacreEOS signature [49]:

STANDARD\_wInit+Cider-D[n4,s6.0]+average[nspis5]+1.0.0

BÀNG I: Nghiên cứu phá hủy trong giai đoạn đầu tiên của quá trình đào tạo Cross-Entropy sử dụng chùm tia có kích thước 3 trên phép chia xác thực Karpathy. B=BLEU. M=METEOR. R=ROUGE. C=CIDEr-D. S=SPICE.

Bộ mã hóa	Bộ giải mã	B1	B2	B3	B4	Ông	C	S
Đường cơ sở	Đường cơ sở	75,3	59,2	45,4	34,6	28,4	57,0	115,8 21,6
Stc. Exp. G={16}	Đường cơ sở	76,4	60,6	46,6	35,5	28,6	57,2	117,8 21,9
Stc. Exp. G={32}	Đường cơ sở	75,9	59,9	46,1	35,2	28,9	57,1	57,1
Stc. Exp. G={64}	Đường cơ sở	76,3	60,4	46,4	35,5	28,8	57,1	22,0
Đường cơ sở	Dyn. Exp. NE=4	77,2	61,4	47,4	36,2	28,9	57,7	119,7 22,3
Đường cơ sở	Dyn. Exp. NE=8	76,9	61,5	47,9	37,1	29,1	Dyn. Exp. NE=16	57,8 120,8 22,3
Đường cơ sở	Dyn. Exp. NE=16	76,7	61,4	47,8	36,8	29,0	57,7	121,2 22,2
Đường cơ sở	Dyn. Exp. NE=16	77,4	61,9	48,2	37,3	29,2	58,0	122,2 22,3
Stc. Exp. G={64}	Dyn. Exp. NE=16	77,8	62,3	48,3	37,2	29,3	58,3	122,8 22,5
Stc. Exp. G={128, 128, 128, 128, 128}	Dyn. Exp. NE=16	77,4	62,0	48,2	37,2	29,2	58,0	122,5 22,2
Stc. Exp. G={256, 256, 256, 256, 256}	Dyn. Exp. NE=16	77,3	61,7	47,9	37,0	29,3	58,1	122,7 22,4
Stc. Exp. G={512, 512, 512, 512, 512}	Dyn. Exp. NE=16	77,6	62,0	48,2	37,2	29,4	58,1	123,5 22,5
Stc. Exp. G={32, 64, 128, 256, 512}	Dyn. Exp. NE=16	77,6	62,0	48,2	37,2	29,4	58,1	

## IV. KẾT QUẢ

### A. Thiết lập thử nghiệm

1) Bộ dữ liệu: Bộ dữ liệu đào tạo bao gồm các dữ liệu phổ biến

Điểm chuẩn MS-COCO [41] chia theo [45], kết quả

trong 113287 cặp hình ảnh mô tả cho việc đào tạo, 5000 trong

bộ xác thực và trong bộ kiểm tra 5000. Mỗi tham chiếu

chú thích được xử lý trước bằng một đường ống đơn giản bao gồm

viết hoa chữ thường, xóa dấu câu và lọc ra các từ

không xuất hiện ít nhất 5 lần (tỷ trọng có kích thước 10000).

Ngoài ra, mô hình cuối cùng được đánh giá qua Novel

Bộ dữ liệu xác thực chú thích đối tượng theo tỷ lệ (nocaps) [46],

bao gồm ba lớp hình ảnh được gọi là trong miền,

gần miền và ngoài miền, theo sự quen thuộc của

các lớp liên quan đến những lớp có trong tập huấn luyện.

Bộ dữ liệu này chịu sự xử lý trước tương tự của MS-COCO và phục vụ mục

dịch tiếp tục thách thức mô hình

trong điều kiện bất lợi.

2) Chi tiết mô hình: Hai mô hình được triển khai cho

thiết lập thử nghiệm. Đường cơ sở, là Base Transformer và mô hình

chính của chúng tôi, được gọi là “ExpansionNet v2”,

được thực hiện với các cấu hình sau  $d_m=512$ ,

$df=2048$ ,  $N_{enc}=N_{dec}=3$ . Trong phần sau, phép mở rộng động

hệ số được đặt thành 16 và hệ số giãn nở tĩnh

gồm  $G=\{32, 64, 128, 256, 512\}$  (chi tiết hơn trong Phần

IV-B). Mỗi cái đều dựa vào định của cùng một xương sống,

Swin-Transformer trong cấu hình lớn [13] được đào tạo trước

trên ImageNet [47]. Tất cả hình ảnh đều phải tuân theo một mức tối thiểu

tiền xử lý: đầu tiên, chúng được thay đổi kích thước thành  $3\times 384\times 384$

tensor, sau đó các giá trị RGB được chuyển đổi thành phạm vi  $[0, 1]$

và được chuẩn hóa thêm bằng cách sử dụng  $mean=(0.485, 0.456, 0.406)$  và

$std=(0.229, 0.224, 0.225)$ . Mã nguồn của các thí nghiệm

có sẵn .

3) Thuật toán đào tạo: Có thể thấy rằng xương sống Swin-Transformer

là xương sống tồn kén nhất về mặt tính toán

một phần của hệ thống. Vì lý do này, lấy cảm hứng từ [48], để

cho phép bước đào tạo End to End đến một số lượng lớn hơn

của kiến trúc tính toán, đào tạo của chúng tôi được chia thành

bốn bước cụ thể, mỗi giai đoạn (trong cả entropy chéo

giai đoạn đào tạo và cung cấp) bao gồm đào tạo ban đầu

trong đó trọng lượng của xương sống được đông lạnh và điều chỉnh tinh vi

bước trong đó các gradient chảy qua toàn bộ hệ thống:

Bước A) Entropy chéo - Xương sống đóng băng. Mô hình là

được đào tạo bằng cách sử dụng kích thước lô 48, tốc độ học ban đầu là  $2e-4$ , a khởi động 10000 và được ủ 0,8 sau mỗi 2 ký nguyên 8 thời đại;

Bước B) Entropy chéo - Từ đầu đến cuối. Toàn bộ hệ thống là

được đào tạo cho 2 ký nguyên bổ sung, sử dụng kích thước lô 48 và một tốc độ học ban đầu là  $3e-5$  được tối ưu bằng 0,55 mỗi ký nguyên;

TABLE II: Offline comparison of State-of-the-Art single models over the Karpathy test split. B=BLEU. M=METEOR. R=ROUGE. C=CIDEr-D. S=SPICE.

Model	Cross-Entropy						CIDEr-D optimization					
	B1	B4	M	R	C	S	B1	B4	M	R	C	S
Up-Down [5]	77.2	36.2	27.0	56.4	113.5	20.3	79.8	36.3	27.7	56.9	120.1	21.4
GCN-LSTM [50]	77.3	36.8	27.9	57.0	116.3	20.9	80.5	38.2	28.5	58.3	127.6	22.0
SGAE [51]	-	-	-	-	-	-	80.8	38.4	28.4	58.6	127.8	22.1
AoANet [31]	77.4	37.2	28.4	57.5	119.8	21.3	80.2	38.9	29.2	58.8	129.8	22.4
X-Transformer [14]	77.3	37.0	28.7	57.5	120.0	21.8	80.9	39.7	29.5	59.1	132.8	23.4
GET [35]	-	-	-	-	-	-	81.5	39.5	29.3	58.9	131.6	22.8
DLCT [18]	-	-	-	-	-	-	81.4	39.8	29.5	59.1	133.8	23.0
RSTNet [52]	-	-	-	-	-	-	81.8	40.1	29.8	59.5	135.6	23.3
PureT [11]	-	-	-	-	-	-	82.1	40.9	30.2	60.1	138.2	24.2
ExpansionNet v2	78.1	38.1	30.1	58.9	128.2	23.5	<b>82.8</b>	<b>41.5</b>	<b>30.3</b>	<b>60.5</b>	<b>140.4</b>	<b>24.5</b>

TABLE III: Offline comparison of State-of-the-Art ensemble models over the Karpathy test split. B=BLEU. M=Meteor. R=Rouge. C=CIDEr-D. S=SPICE.

Model	Cross-Entropy						CIDEr-D optimization					
	B1	B4	M	R	C	S	B1	B4	M	R	C	S
GCN-LSTM [50]	77.4	37.1	28.1	57.2	117.1	21.1	80.9	38.3	28.6	58.5	128.7	22.1
SGAE [51]	-	-	-	-	-	-	81.0	39.0	28.4	58.9	129.1	22.2
AoANet [31]	78.7	38.1	28.5	58.2	122.7	21.7	81.6	40.2	29.3	59.4	132.0	22.8
X-Transformer [14]	77.8	37.7	29.0	58.0	122.1	21.9	81.7	40.7	29.9	59.7	135.3	23.8
GET [35]	-	-	-	-	-	-	82.1	40.6	29.8	59.6	135.1	23.8
DLCT [18]	-	-	-	-	-	-	82.2	40.8	29.9	59.8	137.5	23.3
PureT [11]	-	-	-	-	-	-	83.4	42.1	30.4	60.8	141.0	24.3
ExpansionNet v2	78.5	38.5	29.9	58.8	128.7	23.6	<b>83.5</b>	<b>42.7</b>	<b>30.6</b>	<b>61.1</b>	<b>143.7</b>	<b>24.7</b>

layer in the single group configuration is limited. In fact, it only slightly improves the baseline, regardless of the choice of  $N_E$ . Conversely, the dynamic expansion layer showcases a more significant improvement obtaining the best result for  $N_E = 16$ . When the two expansion methods are combined, the model outperforms the baseline across all metrics with a margin of at least 6.0 CIDEr-D, 2.0 BLEU, 0.5 SPICE, 1.0 ROUGE, and 0.8 METEOR. Analyzing several configurations of length groups in the static expansion, it appears that introducing more expansion vectors does not necessarily lead to better performances, since for  $G = \{128, 128, 128, 128, 128\}$ ,  $G = \{384, 384, 384, 384, 384\}$  and  $G = \{512, 512, 512, 512, 512\}$  the model yield similar results. However, the model seems to benefit from a diverse selection of coefficients such as in the case of  $G = \{32, 64, 128, 256, 512\}$  which will be adopted in the remaining experiments. Ultimately, all instances outperform the baseline across all metrics.

### C. Performance Comparison

1) COCO Offline Evaluation: Table II and Table III report the score comparison between ExpansionNet v2 and the best-performing models in recent years. Up-Down [5] introduced the idea of extracting a collection of features from the images using an object detector like Faster-RCNN [6] in contrast to the classification backbone [4]. The idea was adopted in most of the following architectures as well, for instance, in the case of GCN-LSTM [50] and SGAE [51], which additionally implemented a convolutional graph network on top of it to exploit the information provided by a scene graph. AoANet [31] adopted the Transformer and improved the attentive components with two gates serving the purpose of

simulating an additional level of attention over the inputs and augmented the language modelling part with an LSTM. On the other hand, X-Transformer [14] adopted a fully attentive architecture and further refined the attentive blocks by means of bilinear pooling techniques. The most recent and performing architectures focused on increasing the more effective ways to feed visual information into the sequence modelling network. For instance, RSTNet [52] showcased the effectiveness of grid features over regions, GET [35] processed the images using a global representation in conjunction with the local ones, DLCT [18] instead exploited the advantages of both regions and grid visual features. Finally, PureT [11] implemented the first end-to-end Transformer architecture applying the Window / Shifted-Window MHA [13] in both the encoder and decoder. ExpansionNet v2 outperforms PureT by a margin of 0.7 BLEU1, 0.6 BLEU4, 0.1 METEOR, 0.4 ROUGE, 2.2 CIDEr-D and 0.3 SPICE in the single model case and by 0.1 BLEU1, 0.6 BLEU4, 0.3 ROUGE, 2.7 CIDEr-D and 0.4 SPICE in the ensemble configuration.

2) COCO Online Evaluation: We evaluate ExpansionNet v2 using the ensemble configuration and adopting the standard Beam Search (beam size 5) over the official testing set of 40775 images, submitting the predictions to the online testing server. Results are reported in Table IV. c5 and c40 represent the scores of 5 and 40 reference captions (unknown to the user), respectively.

Our model achieves State-of-the-Art performance (as of 2 July 2022) among non-generative models trained on MS-COCO 2014, outperforming the previous one [11] by a margin of 1.2 BLEU4 (c40), 0.2 METEOR (c40), 0.5 ROUGE-L (c40)

BÀNG II: So sánh ngoại tuyến các mô hình đơn lẻ hiện đại qua phép chia tách thử nghiệm Karpathy. B=BLEU. M=METEOR. R=ĐỎ. C=RƯỢU CỐC. S=GIA VI.

Mô hình	Entropy chéo						Tối ưu hóa CIDEr-D					
	B1	B4	MRC	S			B1	B4	MR	C	S	
Lên-Xuống [5]	77,2	36,2	27,0	56,4	113,5	20,3	79,8	36,3	27,7	56,9	120,1	21,4
GCN-LSTM [50]	77,3	36,8	27,9	57,0	116,3	20,9	80,5	38,2	28,5	58,3	127,6	22,0
SGAE [51]	-	-	-	-	-	-	-	-	-	80,8	38,4	28,4
AoANet [31]	77,4	37,4	37,2	28,4	57,5	X-Transformer [14]	119,8	21,3	80,2	38,9	29,2	25,8
77,3	37,0	28,7	57,5	NHÂN	[35]	-	120,0	21,8	80,9	39,7	29,5	23,4
DLCT [18]	-	-	-	-	-	-	-	-	-	81,5	39,5	29,3
Mạng RST [52]	-	-	-	-	-	-	-	-	-	81,4	39,8	29,5
PureT [11]	-	-	-	-	-	-	-	-	-	81,8	40,1	29,8
Mở rộngNet v2	78,1	38,1	30,1	58,9	128,2	-	128,2	23,5	82,8	41,5	30,3	60,5

BÀNG III: So sánh ngoại tuyến các mô hình tổng hợp hiện đại qua phép chia tách thử nghiệm Karpathy. B=BLEU. M=Meteor. R=ĐỎ. C=Rượu táo-D. S=GIA VI.

Người mẫu	Entropy chéo						Tối ưu hóa CIDEr-D					
	B1	B4	MRC	S			B1	B4	MR	C	S	
GCN-LSTM [50]	77,4	37,1	28,1	57,2	117,1	21,1	80,9	38,6	58,6	5,128,7	22,1	
SGAE [51]	-	-	-	-	-	-	-	-	-	81,0	39,0	28,4
Mạng AoA [31]	78,7	38,1	28,5	58,2	122,7	21,7	81,6	40,2	29,3	59,4	132,0	22,8
Máy biến áp X [14]	77,8	37,7	29,0	58,0	122,1	21,9	81,7	40,7	29,9	59,7	135,3	23,8
NHÂN [35]	-	-	-	-	-	-	-	-	-	82,1	40,6	29,8
DLCT [18]	-	-	-	-	-	-	-	-	-	82,2	40,8	29,9
PureT [11]	-	-	-	-	-	-	-	-	-	83,4	42,1	30,4
Mở rộngNet v2	78,5	38,5	29,9	58,8	-	-	128,7	23,6	83,5	42,7	30,6	61,1

lớp trong cấu hình nhóm đơn bị hạn chế. Trên thực tế, nó chỉ cải thiện một chút đường cơ sở, bắt kể sự lựa chọn của NE. Ngược lại, lớp mở rộng động thể hiện một cải tiến đáng kể hơn đạt được kết quả tốt nhất cho NE = 16. Khi hai phương pháp mở rộng được kết hợp, mô hình vượt trội hơn đường cơ sở trên tất cả các số liệu với biên độ ít nhất là 6.0 CIDEr-D, 2.0 BLEU, 0.5 SPICE, 1.0 ROUGE và 0.8 METEOR. Phân tích một số cấu hình của các nhóm chiều dài trong sự mở rộng tĩnh, có vẻ như việc giới

TABLE IV: Online server results on the MS-COCO 2014 test set which ground truth is unknown. B=BLEU. M=METEOR. R=ROUGE. C=CIDEr-D. S=SPICE.

Model	B1		B2		B3		B4		METEOR		ROUGE-L		CIDEr-D	
	c5	c40	c5	c40	c5	c40	c5	c40	c5	c40	c5	c40	c5	c40
Up-Down [5]	80.2	95.2	64.1	88.8	49.1	79.4	36.9	68.5	27.6	36.7	57.1	72.4	117.9	120.5
GCN-LSTM [50]	-	-	65.5	89.3	50.8	80.3	38.7	69.7	28.5	37.6	58.5	73.4	125.3	126.5
SGAE [51]	81.0	95.3	65.6	89.5	50.7	80.4	38.5	69.7	28.2	37.2	58.6	73.6	123.8	126.5
AoANet [31]	81.0	95.0	65.8	89.6	51.4	81.3	39.4	71.2	29.1	38.5	58.9	74.5	126.9	129.6
X-Transformer [14]	81.9	95.7	66.9	90.5	52.4	82.5	40.3	72.4	29.6	39.2	59.5	75.0	131.1	133.5
RSTNet [52]	82.1	96.4	67.0	91.3	52.2	83.0	40.0	73.1	29.6	39.1	59.5	74.6	131.9	134.0
GET [35]	81.6	96.1	66.5	90.9	51.9	82.8	39.7	72.9	29.4	38.8	59.1	74.4	130.3	132.5
DLCT [18]	82.4	96.6	67.4	91.7	52.8	83.8	40.6	74.0	29.8	39.6	59.8	75.3	133.3	135.4
PureT [11]	82.8	96.5	68.1	91.8	53.6	83.9	41.4	74.1	30.1	39.9	60.4	75.9	136.0	138.3
OFA [15]	<b>84.5</b>	<b>98.1</b>	70.1	<b>94.4</b>	<b>55.9</b>	<b>87.8</b>	<b>43.6</b>	<b>78.7</b>	<b>32.1</b>	<b>42.7</b>	<b>62.5</b>	<b>79.0</b>	<b>147.2</b>	149.6
GIT [16]	84.3	<b>98.1</b>	70.0	<b>94.4</b>	55.7	87.6	43.2	78.3	31.9	42.1	62.0	78.4	146.4	<b>149.8</b>
ExpansionNet v2	83.3	96.9	8.8	92.6	54.4	85.0	42.1	75.3	30.4	40.1	60.8	76.4	138.5	140.8

and 2.5 CIDEr-D in both c5 and c40 instances. However, it is ultimately outperformed, by a significant margin, by generative models [15], [16] which we consider orthogonal to our work since they focus more on training method and data quality rather than architecture design.

3) *Nocaps Evaluation*: We evaluate ExpansionNet v2 over the nocaps validation set. In particular, we adopt a single model trained exclusively on Cross-Entropy Loss, using no additional pre-training data sets. The predictions are generated by the standard Beam Search algorithm (beam size 3) in contrast to the CBS [54]. A limited comparison is reported in Table V, which showcases that our model achieves very competitive results among the architectures trained in similar configurations, with an overall lead of 17.6 CIDEr and 1.4 SPICE over the Up-Down model [5]. It is still ultimately outperformed by recent V+L pre-training-based works such

TABLE V: Performances on nocaps validation set. C and S denote the CIDEr-D and SPICE scores respectively.

Domain	Metric	Enc-Dec [55]	Up-Down [5]	Ours
In	C	72.8	78.1	<b>83.8</b>
	S	11.1	11.6	<b>12.6</b>
Near	C	57.1	57.7	<b>79.2</b>
	S	10.2	10.3	<b>12.4</b>
Out	C	34.1	31.3	<b>54.0</b>
	S	8.3	8.3	<b>9.3</b>
All	C	54.7	55.3	<b>72.9</b>
	S	10.0	10.1	<b>11.4</b>

estimated assuming all model computational costs are the same as the ExpansionNet v2, which is a generous approximation compared to generative models whose sizes are tens of times larger. Despite such premise and the fact that we also perform end-to-end training, it can be seen that our model can be trained up to  $2.8\times$  faster than other non-generative models and up to  $46.8\times$  faster in the case of generative ones. Recalling the results in Table IV, performance-wise, our model achieves 93.9% performances of the State-of-the-Art model GIT [16] but uses 7080× less data and is 129× smaller.

TABLE VI: Inference cost comparison of ablation models on the MS-COCO 2014 validation set (5000 images).

Encoder	Decoder	FLOPS
Baseline	Baseline	$9.28 \times 10^{12}$
Stc. Exp. G={16}	Baseline	$9.62 \times 10^{12}$
Stc. Exp. G={32}	Baseline	$9.70 \times 10^{12}$
Stc. Exp. G={64}	Baseline	$9.88 \times 10^{12}$
Baseline	Dyn. Exp. $N_E=4$	$9.40 \times 10^{12}$
Baseline	Dyn. Exp. $N_E=8$	$9.43 \times 10^{12}$
Baseline	Dyn. Exp. $N_E=16$	$9.48 \times 10^{12}$
Stc. Exp. G={64}	Dyn. Exp. $N_E=16$	$10.08 \times 10^{12}$
Stc. Exp. G={128}×5	Dyn. Exp. $N_E=16$	$13.26 \times 10^{12}$
Stc. Exp. G={256}×5	Dyn. Exp. $N_E=16$	$16.80 \times 10^{12}$
Stc. Exp. G={512}×5	Dyn. Exp. $N_E=16$	$23.88 \times 10^{12}$
ExpansionNet v2	ExpansionNet v2	$15.21 \times 10^{12}$

### E. Qualitative Analysis

Table VIII provides some examples of captions. Regardless of the image complexity, ExpansionNet v2 is not only able to correctly describe the subjects depicted in the scenes but also showcases a good level of semantic understanding by describing the goals and interactions. Unfortunately, our model seems to struggle with out-of-domain objects as showcased in Table IX where, due to objects and terms unknown to the model, predictions are either imprecise (2<sup>nd</sup> image) or incorrect (1<sup>st</sup> image). Nonetheless, it appears to provide a roughly correct description of the image. We showcase an example of attention visualization in Fig. 4, where the scattered focus correctly outlines the main subjects despite the absence of an object detector.

as [39], [40], [56].

### D. Training and Inference Cost

The efficiency aspect was addressed in the design of the Expansion mechanism. For instance, it can be observed from Table VI that doubling the expansion coefficients does not lead to double FLOPS, which would be the case of actually doubling the input sequence length. In particular, for small parameters, our model is comparable to the Transformer in terms of computational cost. In contrast, ExpansionNet v2 is  $1.63\times$  slower than the baseline because of an abundant selection of expansion coefficients.

In Table VII, we compare our training time with the ones presented by other works. In particular, time entries are

BÀNG IV: Kết quả máy chủ trực tuyến trên bộ thử nghiệm MS-COCO 2014 mà sự thật cơ bản chưa được biết. B=BLEU. M=METEOR. R=ROUGE. C=RƯỢU CỐC. S=GIA VI.

Người mẫu	B1		B2		B3		B4		METEOR-L ĐỎ		CIDEr-D	
	c5	c40	c5	c40	c5	c40	c5	c40	c5	c40	c5	c40
Lên-Xuống [5]	80,2	95,2	64,1		88,8	49,1	79,4	36,9	68,5	27,6	36,7	57,1
GCN-LSTM [50]	-	-	65,5	89,3	50,8	80,3	38,7	69,7	28,5	37,6	58,5	73,4
SGAE [51]	81,0	95,3	65,6	89,5	50,7	80,4	38,5	69,7	28,2	37,2	58,6	73,6
Mạng AoA [31]	81,0	95,0	65,8	89,6	51,4	81,3	39,4	71,2	29,1	38,5	58,9	74,5
X-Transformer [14]	81,9	95,7	66,9	90,5	52,4	82,5	40,3	72,4	29,6	39,2	59,5	75,0
RSTNet [52]	82,1	96,4	67,0	91,3	52,2	83,0	40,0	73,1	29,6	39,1	59,5	74,6
GET [35]	81,6	96,1	66,5	90,9	51,9	82,8	39,7	72,9	29,4	38,8	59,1	74,4
DLCT [18]	82,4	96,6	67,4	91,7	52,8	83,8	40,6	74,0	29,8	39,6	59,8	75,3
PureT [11]	82,8	96,5	68,1	91,8	53,6	83,9	41,4	74,1	30,1	39,9	60,4	75,9
OFA [15]	<b>84,5</b>	<b>98,1</b>	70,1	<b>94,4</b>	<b>55,9</b>	<b>87,8</b>	<b>43,6</b>	<b>78,7</b>	<b>32,1</b>	<b>42,7</b>	<b>62,5</b>	<b>79,0</b>
GIT [16]	84,3	<b>98,1</b>	70,0	<b>94,4</b>	55,7	87,6	43,2	78,3	31,9	42,1	62,0	78,4
ExpansionNet v2	83,3	96,9	8,8	92,6	54,4	85,0	42,1	75,3	30,4	40,1	60,8	76,4
Mở rộngNet v2	83,3	96,9	92,6	54,8	48,8	85,0	42,1	75,3	30,4	40,1	60,8	76,4

và 2,5 CIDEr-D trong cả hai trường hợp c5 và c40. Tuy nhiên, nó là cuối cùng đã vượt trội hơn, với biên độ đáng kể, bởi khả năng tạo ra các mô hình [15], [16] mà

TABLE VII: Training time comparison of State-of-the-Art works against our solution. “Time” represents the estimated time required to train models on a single NVIDIA A100 using the described strategy.  $\gamma$ ,  $\theta$  and  $\sigma$  denote the normalized quantity of the number of parameters, the number of training images and the training cost compared to our proposal. The “★” symbol denotes generative modes, typically pre-trained on multiple tasks and images from various sources. We simplify the matter using the cost of Cross-Entropy training on MS-COCO 2014, and the downstream task learning cost is ignored since it is negligible compared to the pre-training phase.

Source	Params. ( $\gamma$ )	Datasets → total num. images ( $\theta$ )	Training Description	Train. time ( $\sigma$ )
Obj. Transf. [17], AoANet [31]	33M (0.86)	MS-COCO 2014 → 113K (1.00)	Cross-Entropy: ~ 30 epochs and batch size 10. Reinforcement: 30 epochs and batch size 10.	7 days (2.80)
PureT [11]	87M (2.28)			
X-Transformer [14]	34M (0.89)	MS-COCO 2014 → 113K (1.00)	Cross-Entropy: 70 epochs and batch size 40. Reinforcement: 35 epochs and batch size 32.	5 days (2.00)
GIT [16] *	4.9B (128.94)	MS-COCO, CC3M, CC12M, VG, SBG, ALT200M + 0.6B → 0.8B (7079.64).	2 epochs and we assume batch size 48 in the estimation.	117 days (46.80)
OFA [15] *	871M (22.92)	MS-COCO, CC3M, CC12M, VG, SBG → 15M (132.74).	40 epochs and assume batch size 48 in the estimation.	44 days (17.60)
ExpansionNet v2	38M (1.00)	MS-COCO 2014 → 113K (1.00)	See Section IV-A.	2.5 days (1.00)

TABLE VIII: Examples of captions.

Image	Captions
	<b>Baseline:</b> A man holding a tennis ball on a tennis court. <b>ExpansionNet v2:</b> A man jumping in the air to hit a tennis ball. <b>Gt:</b> {A tennis player jumps and swats at the ball.; A tennis player hitting a tennis ball on a court.; Professional tennis player immediately after returning a shot.}
	<b>Baseline:</b> A little girl brushing her hair with a table. <b>ExpansionNet v2:</b> A little girl brushing her hair with a pink brush. <b>Gt:</b> {A young girl tries to comb her own hair.; A young child brushing her hair with a big pink brush.; A young girl is trying to brush her hair with a pink brush.}

## V. CONCLUSION

In this work, we addressed the question of whether the fixed number of elements of the inputs represented a performance bottleneck in modern image-captioning systems. To this end, we presented the idea of an Expansion mechanism and provided two concrete implementations called Static Expansion and Dynamic Expansion, that process the input using sequences that feature a different length compared to the one provided in the input. Upon these layers, we designed a new architecture called ExpansionNet v2 and trained it on the MS-COCO 2014 dataset using a fast End to End training approach. Extensive experiments conducted on the testing set showcase that our method achieved better performances when compared to the baseline. This answer positively the initial research question of whether the input length can represent a bottleneck to the sequence processing. Additionally, ExpansionNet v2 achieved strong performances on both offline (143.7 CIDEr-D) and online (140.8 CIDEr-D) test splits and is outperformed mainly by V+L pre-training models, which we consider or-

thogonal to our work due to the differences in model size and additional training data. In conclusion, we introduced the Expansion layers and ExpansionNet v2 and found the answer to our research question in the case of the Image Captioning field. Future works will further develop the methods and ideas presented in this work, motivated by the fact that they can be easily integrated into other solution approaches (such as V+L pre-training) and other research fields.

TABLE IX: Examples on nocaps out-of-domain images.

Image	Captions
	<b>ExpansionNet v2:</b> A close-up of a fish in a body of water. <b>3 gts:</b> { A seahorse in an aquarium full of water with some plants growing in the background.; A blue seahorse is swimming near sea plants on back.; A very small seahorse is in the water along with other pieces. }
	<b>ExpansionNet v2:</b> Three pictures of a blender with red liquid in it. <b>3 gts:</b> { A picture of three blenders with a strawberry looking beverage inside.; A white mixer in the process of making a smoothie.; The steps of making a smoothie in a blender are shown. }

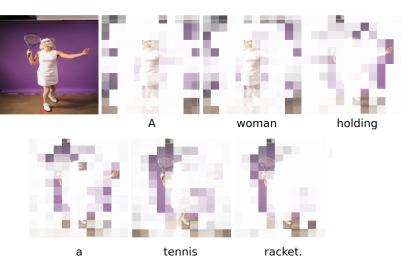


Fig. 4: Attention visualization of a single decoder head in ExpansionNet v2.

BẢNG VII: So sánh thời gian đào tạo của State-of-the-Art so với giải pháp của chúng tôi. “Thời gian” biểu thị thời gian ước tính cần thiết để đào tạo các mô hình trên một NVIDIA A100 duy nhất bằng chiến lược được mô tả.  $\gamma$ ,  $\theta$  và  $\sigma$  là số lượng chuẩn hóa của số lượng tham số, số lượng hình ảnh đào tạo và chi phí đào tạo so với đề xuất của chúng tôi. Ký hiệu “\*” biểu thị các chế độ đào tạo, thường được đào tạo trước trên nhiều tác vụ và hình ảnh từ nhiều nguồn khác nhau. Chúng tôi đơn giản hóa vấn đề bằng cách sử dụng chi phí đào tạo Cross-Entropy trên MS-COCO 2014 và chi phí học tác vụ hạ lưu được bỏ qua vì nó không đáng kể so với giai đoạn đào tạo trước.

Nguồn	Tham số ( $\gamma$ )	Bộ dữ liệu	tổng số hình ảnh ( $\theta$ )	Mô tả đào tạo Cross-Entropy:	Xe lửa, thời
đối tượng Chuyển khoản. [17], AoANet [31]	33M(0,86)	MS-COCO 2014	113K (1.00)	30 ký nguyên và quy mô lô 10. Cùng với: 30 ký nguyên và quy mô lô 10.	gian 7 ngày (2,80)
PureT [11]	87 triệu (2,28)				
X-Transformer [14]	34M (0,89)	MS-COCO 2014	113K (1.00)	Cross-Entropy: 70 ký nguyên và kích thước lô là 40. Reinforcement: 35 ký nguyên và kích thước lô là 32.	5 ngày (2.00)
GIT [16]	141M (3,71)			2 ký nguyên và chúng tôi giả định kích thước lô là 48 trong ước tính.	117 ngày (46,80)
OFA [15]	4.9B (128.94)	MS-COCO, CC3M, CC12M, VG, SBG, ALT200M + 0.6B → 0.8B (7079.64).		40 ký nguyên và giả sử kích thước lô là 48 trong ước tính.	44 ngày (17,60)
ExpansionNet v2	871M (22.92)	MS-COCO, CC3M, CC12M, VG, SBG → 15M (132.74).		Xem Phần IV-A.	2,5 ngày (1,00)

BẢNG VIII: Ví dụ về chú thích.

Hình ảnh	Chú thích
	Đường cơ sở: Một người đàn ông đang cầm một quả bóng tennis trên sân tennis. ExpansionNet v2: Một người đàn ông nhảy lên không trung để đánh một quả bóng tennis. Gt: {Một vận động viên quần vợt nhảy lên và đánh vào quả bóng.; Một vận động viên quần vợt đánh một quả bóng quần vợt trên sân.; Một vận động viên quần vợt chuyên nghiệp ngay sau khi trả bóng.}
	Đường cơ sở: Một bé gái đang chải tóc bằng bàn. ExpansionNet v2: Một bé gái đang chải tóc bằng chiếc lược màu hồng. Gt: {Một cô gái trẻ đang cố gắng tự chải tóc của mình.; Một đứa trẻ đang chải tóc bằng một chiếc lược lớn màu hồng.; Một cô gái trẻ đang cố gắng chải tóc bằng một chiếc lược màu hồng.}

## V. KẾT LUẬN

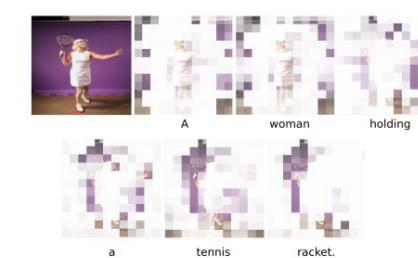
Trong công trình này, chúng tôi giải quyết câu hỏi liệu số lượng phần tử cố định của đầu vào có phải là nút thắt cổ chai về hiệu suất trong các hệ thống chủ thích hình ảnh hiện đại hay không. Để đạt được mục đích này, chúng tôi đã trình bày ý tưởng về cơ chế Mở rộng và cung cấp hai triển khai cụ thể được gọi là Mở rộng tĩnh và Mở rộng động, xử lý đầu vào bằng các chuỗi có độ dài khác nhau với chuỗi được cung cấp trong đầu vào. Trên các lớp này, chúng tôi đã thiết kế một kiến trúc mới có tên là ExpansionNet v2 và đào tạo kiến trúc này trên tập dữ liệu MS-COCO 2014 bằng cách sử dụng phương pháp đào tạo End to End nhanh. Các thí nghiệm mở rộng được tiến hành trên bộ thử nghiệm cho thấy phương pháp của chúng tôi đạt được hiệu suất tốt hơn khi so sánh với đường cơ sở. Câu trả lời tích cực này trả lời câu hỏi nghiên cứu ban đầu về việc liệu độ dài đầu vào có thể là nút thắt cổ chai đối với quá trình xử lý chuỗi hay không. Ngoài ra, ExpansionNet v2 đạt được hiệu suất mạnh mẽ trên cả phân tách thử nghiệm ngoại tuyến (143,7 CIDEr-D) và trực tuyến (140,8 CIDEr-D) và chủ yếu bị vượt trội bởi các mô hình tiền huấn luyện V+L, mà chúng tôi coi là hoặc-

thogonal cho công việc của chúng tôi do sự khác biệt về kích thước mô hình và dữ liệu đào tạo bổ sung. Tóm lại, chúng tôi đã giới thiệu các lớp Expansion và ExpansionNet v2 và tìm thấy câu trả lời cho câu hỏi nghiên cứu của chúng tôi trong trường hợp lĩnh vực Chủ thích hình ảnh. Các công trình trong tương lai sẽ tiếp tục phát triển các phương pháp và ý tưởng được trình bày trong công trình này, được thúc đẩy bởi thực tế là chúng có thể dễ dàng tích hợp vào các phương pháp giải pháp khác (như tiền đào tạo V+L) và các lĩnh vực nghiên cứu khác.

BẢNG IX: Ví dụ về hình ảnh ngoài miền nocaps.

Hình ảnh	Chú thích
	ExpansionNet v2: Cảnh cảnh một con cá trong một vùng nước. 3 gts: { Một con cá ngựa trong một bể đầy nước với một số cây mọc ở phía sau.; Một con cá ngựa xanh đang bơi gần những cây biển ở phía sau.; Một con cá ngựa xanh nhỏ đang ở trong nước cùng với những mảnh khác. }

ExpansionNet v2: Ba hình ảnh về máy xay sinh tố có chứa chất lỏng màu đỏ. 3 gts: { Hình ảnh về ba máy xay sinh tố có chứa đồ uống trông giống quả dưa tây bên trong.; Một con cá ngựa xanh đang bơi gần những cây biển ở phía sau.; Các bước làm sinh tố trong máy xay sinh tố được hiển thị. }



Hình 4: Hình ảnh trực quan về một đầu giải mã duy nhất trong ExpansionNet v2.

## REFERENCES

- [1] M. Mitchell *et al.*, "Midge: Generating image descriptions from computer vision detections," in *Proceedings of the 13th Conference of the European Chapter of the Association for Computational Linguistics*, 2012, pp. 747–756.
- [2] G. Kulkarni *et al.*, "Babytalk: Understanding and generating simple image descriptions," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 35, no. 12, pp. 2891–2903, 2013.
- [3] O. Vinyals, A. Toshev, S. Bengio, and D. Erhan, "Show and tell: A neural image caption generator," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2015, pp. 3156–3164.
- [4] K. Xu *et al.*, "Show, attend and tell: Neural image caption generation with visual attention," in *International conference on machine learning*, 2015, pp. 2048–2057.
- [5] P. Anderson *et al.*, "Bottom-up and top-down attention for image captioning and visual question answering," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2018, pp. 6077–6086.
- [6] S. Ren, K. He, R. Girshick, and J. Sun, "Faster r-cnn: Towards real-time object detection with region proposal networks," *arXiv preprint arXiv:1506.01497*, 2015.
- [7] S. Hochreiter and J. Schmidhuber, "Long short-term memory," *Neural computation*, vol. 9, no. 8, pp. 1735–1780, 1997.
- [8] K. Cho *et al.*, "Learning phrase representations using rnn encoder-decoder for statistical machine translation," *arXiv preprint arXiv:1406.1078*, 2014.
- [9] D. Bahdanau, K. Cho, and Y. Bengio, "Neural machine translation by jointly learning to align and translate," *arXiv preprint arXiv:1409.0473*, 2014.
- [10] A. Vaswani *et al.*, "Attention is all you need," in *Advances in neural information processing systems*, 2017, pp. 5998–6008.
- [11] Y. Wang, J. Xu, and Y. Sun, "End-to-end transformer based model for image captioning," *arXiv preprint arXiv:2203.15350*, 2022.
- [12] V.-Q. Nguyen, M. Suganuma, and T. Okatani, "Grit: Faster and better image captioning transformer using dual visual features," in *Computer Vision–ECCV 2022: 17th European Conference, Tel Aviv, Israel, October 23–27, 2022, Proceedings, Part XXXVI*. Springer, 2022, pp. 167–184.
- [13] Z. Liu *et al.*, "Swin transformer: Hierarchical vision transformer using shifted windows," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2021, pp. 10012–10 022.
- [14] Y. Pan, T. Yao, Y. Li, and T. Mei, "X-linear attention networks for image captioning," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2020, pp. 10 971–10 980.
- [15] P. Wang *et al.*, "Ofa: Unifying architectures, tasks, and modalities through a simple sequence-to-sequence learning framework," in *International Conference on Machine Learning*. PMLR, 2022, pp. 23 318–23 340.
- [16] J. Wang *et al.*, "Git: A generative image-to-text transformer for vision and language," *arXiv preprint arXiv:2205.14100*, 2022.
- [17] S. Herdade, A. Kappler, K. Boakye, and J. Soares, "Image captioning: Transforming objects into words," *arXiv preprint arXiv:1906.05963*, 2019.
- [18] Y. Luo *et al.*, "Dual-level collaborative transformer for image captioning," in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 35, no. 3, 2021, pp. 2286–2293.
- [19] Q. Guo, X. Qiu, P. Liu, Y. Shao, X. Xue, and Z. Zhang, "Star-transformer," *arXiv preprint arXiv:1902.09113*, 2019.
- [20] J. Hao, X. Wang, B. Yang, L. Wang, J. Zhang, and Z. Tu, "Modeling recurrence for transformer," *arXiv preprint arXiv:1904.03092*, 2019.
- [21] M.-T. Luong, H. Pham, and C. D. Manning, "Effective approaches to attention-based neural machine translation," *arXiv preprint arXiv:1508.04025*, 2015.
- [22] A. Raganato, Y. Scherrer, and J. Tiedemann, "Fixed encoder self-attention patterns in transformer-based machine translation," *arXiv preprint arXiv:2002.10260*, 2020.
- [23] Y. Tay, D. Bahri, D.-C. Juan, Z. Zhao, and C. Zheng, "Synthesizer: Rethinking self-attention for transformer models," in *International conference on machine learning*. PMLR, 2021, pp. 10 183–10 192.
- [24] H. Ramsauer *et al.*, "Hopfield networks is all you need," *arXiv preprint arXiv:2008.02217*, 2020.
- [25] W. You, S. Sun, and M. Iyyer, "Hard-coded gaussian attention for neural machine translation," *arXiv preprint arXiv:2005.00742*, 2020.
- [26] J. Lee-Thorp, J. Ainslie, I. Eckstein, and S. Ontanon, "Fnet: Mixing tokens with fourier transforms," *arXiv preprint arXiv:2105.03824*, 2021.
- [27] I. O. Tolstikhin *et al.*, "Mlp-mixer: An all-mlp architecture for vision," *Advances in neural information processing systems*, vol. 34, pp. 24 261–24 272, 2021.
- [28] R. Socher and L. Fei-Fei, "Connecting modalities: Semi-supervised segmentation and annotation of images using unaligned text corpora," in *2010 IEEE Computer Society Conference on Computer Vision and Pattern Recognition*. IEEE, 2010, pp. 966–973.
- [29] B. Z. Yao, X. Yang, L. Lin, M. W. Lee, and S.-C. Zhu, "I2t: Image parsing to text description," *Proceedings of the IEEE*, vol. 98, no. 8, pp. 1485–1508, 2010.
- [30] L. Wang, Z. Bai, Y. Zhang, and H. Lu, "Show, recall, and tell: Image captioning with recall mechanism," in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 34, no. 07, 2020, pp. 12 176–12 183.
- [31] L. Huang, W. Wang, J. Chen, and X.-Y. Wei, "Attention on attention for image captioning," in *Proceedings of the IEEE International Conference on Computer Vision*, 2019, pp. 4634–4643.
- [32] J.-H. Kim, J. Jun, and B.-T. Zhang, "Bilinear attention networks," *arXiv preprint arXiv:1805.07932*, 2018.
- [33] S. Sukhbaatar, E. Grave, G. Lample, H. Jegou, and A. Joulin, "Augmenting self-attention with persistent memory," *arXiv preprint arXiv:1907.01470*, 2019.
- [34] A. Gulati *et al.*, "Conformer: Convolution-augmented transformer for speech recognition," *arXiv preprint arXiv:2005.08100*, 2020.
- [35] J. Ji *et al.*, "Improving image captioning by leveraging intra-and inter-layer global representation in transformer network," in *Proceedings of the AAAI conference on artificial intelligence*, vol. 35, no. 2, 2021, pp. 1655–1663.
- [36] M. Cornia, M. Stefanini, L. Baraldi, and R. Cucchiara, "Meshed-memory transformer for image captioning," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2020, pp. 10 578–10 587.
- [37] P. Zeng, H. Zhang, J. Song, and L. Gao, "S2 transformer for image captioning," in *Proceedings of the International Joint Conferences on Artificial Intelligence*, vol. 5, 2022.
- [38] P. Wang *et al.*, "Unifying architectures, tasks, and modalities through a simple sequence-to-sequence learning framework," *arXiv preprint arXiv:2202.03052*, 2022.
- [39] X. Hu *et al.*, "Scaling up vision-language pre-training for image captioning," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2022, pp. 17 980–17 989.
- [40] J. Li, D. Li, C. Xiong, and S. Hoi, "Blip: Bootstrapping language-image pre-training for unified vision-language understanding and generation," *arXiv preprint arXiv:2201.12086*, 2022.
- [41] T.-Y. Lin *et al.*, "Microsoft coco: Common objects in context," in *European conference on computer vision*. Springer, 2014, pp. 740–755.
- [42] R. Xiong *et al.*, "On layer normalization in the transformer architecture," in *International Conference on Machine Learning*. PMLR, 2020, pp. 10 524–10 533.
- [43] S. J. Rennie, E. Marcheret, Y. Mroueh, J. Ross, and V. Goel, "Self-critical sequence training for image captioning," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2017, pp. 7008–7024.
- [44] R. Luo, "A better variant of self-critical sequence training," *arXiv preprint arXiv:2003.09971*, 2020.
- [45] A. Karpathy and L. Fei-Fei, "Deep visual-semantic alignments for generating image descriptions," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2015, pp. 3128–3137.
- [46] H. Agrawal *et al.*, "Nocaps: Novel object captioning at scale," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2019, pp. 8948–8957.
- [47] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei, "Imagenet: A large-scale hierarchical image database," in *2009 IEEE conference on computer vision and pattern recognition*. Ieee, 2009, pp. 248–255.
- [48] J. C. Hu, R. Cavigchioli, and A. Capotondi, "Exploring the sequence length bottleneck in the transformer for image captioning," 2022.
- [49] J. Hu, R. Cavigchioli, and A. Capotondi, "A request for clarity over the end of sequence token in the self-critical sequence training," in *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, vol. 14233 LNCS, p. 39 – 50, 2023.

## TÀI LIỆU THAM KHẢO

- [1] M. Mitchell và cộng sự, "Midge: Tạo mô tả hình ảnh từ các phát hiện thi giác máy tính," trong Biên bản Hội nghị lần thứ 13 của Chi hội Châu Âu thuộc Hiệp hội Ngôn ngữ học tính toán, 2012, trang 747–756.
- [2] G. Kulkarni *et al.*, "Babytalk: Hiểu và tạo ra các mô tả hình ảnh đơn giản," Giao dịch IEEE về Phân tích mẫu và Trí tuệ máy móc, tập 35, số 12, trang 2891–2903, 2013.
- [3] O. Vinyals, A. Toshev, S. Bengio và D. Erhan, "Trình bày và kể: Một trình tạo chú thích hình ảnh thần kinh," trong Biên bản hội nghị IEEE về thi giác máy tính và nhận dạng mẫu, 2015, trang 3156–3164.
- [4] K. Xu *et al.*, "Hiển thị, tham dự và kể: Tạo chú thích hình ảnh bằng nơ-ron với sự chú ý trực quan," trong Hội nghị quốc tế về máy học, 2015, tr. 2048–2057.
- [5] P. Anderson *et al.*, "Sử chú ý từ dưới lên và từ trên xuống đối với chú thích hình ảnh và trả lời câu hỏi trực quan," trong Biên bản báo cáo hội nghị IEEE về thi giác máy tính và nhận dạng mẫu, 2018, tr. 6077–6086.
- [6] S. Ren, K. He, R. Girshick và J. Sun, "R-cnn nhanh hơn: Hướng tới phát hiện đối tượng theo thời gian thực với mạng đẽ xuất vùng," bản in trước arXiv arXiv:1506.01497, 2015.
- [7] S. Hochreiter và J. Schmidhuber, "Bộ nhớ dài hạn ngắn hạn," Thần kinh tính toán, tập 9, số 8, trang 1735–1780, 1997.
- [8] K. Cho *et al.*, "Học cách biểu diễn cụm từ bằng bộ mã hóa-giải mã nnn cho dịch máy thông kê," bản in trước arXiv arXiv:1406.1078, 2014.
- [9] D. Bahdanau, K. Cho và Y. Bengio, "Dịch máy thần kinh bằng cách học chung để cân chỉnh và dịch," bản in trước arXiv arXiv:1409.0473, 2014.
- [10] A. Vaswani và cộng sự, "Sự chú ý là tất cả những gì bạn cần," trong Những tiến bộ trong hệ thống xử lý thông tin thần kinh, 2017, trang 5998–6008.
- [11] Y. Wang, J. Xu và Y. Sun, "Mô hình dựa trên bộ biến đổi đầu cuối để chú thích hình ảnh," bản in trước arXiv arXiv:2203.15350, 2022.
- [12] V.-Q. Nguyen, M. Suganuma và T. Okatani, "Grit: Bộ chuyển đổi chú thích hình ảnh nhanh hơn và tốt hơn bằng cách sử dụng các tính năng trực quan kép," trong Computer Vision-ECCV 2022: Hội nghị Châu Âu lần thứ 17, Tel Aviv, Israel, ngày 23–27 tháng 10 năm 2022, Biên bản báo cáo, Phần XXXVI. Springer, 2022, trang 167–184.
- [13] Z. Liu *et al.*, "Máy biến áp Swin: Máy biến áp tầm nhìn phân cấp sử dụng cửa sổ dịch chuyển," trong Biên bản Hội nghị quốc tế IEEE/CVF về Tầm nhìn máy tính, 2021, trang 10 012–10 022.
- [14] Y. Pan, T. Yao, Y. Li và T. Mei, "Mạng chú ý tuyên tính X để chú thích hình ảnh," trong Biên bản Hội nghị IEEE/CVF về Thị giác máy tính và Nhận dạng mẫu, 2020, trang 10 971–10 980.
- [15] P. Wang *et al.*, "Ofa: Thông nhất các kiến trúc, nhiệm vụ và phương thức thông qua một khuôn khổ học tập trình tự sang trình tự đơn giản," bản in trước arXiv arXiv:2202.03052, 2022.
- [16] X. Hu *et al.*, "Mô rộng quy mô đào tạo trước ngôn ngữ thi giác để chú thích hình ảnh," trong Biên bản Hội nghị IEEE/CVF về Thị giác máy tính và Nhận dạng mẫu, 2022, trang 17 980–17 989.
- [17] J. Li, D. Li, C. Xiong và S. Hoi, "Blip: Khởi động quá trình đào tạo trước ngôn ngữ-hình ảnh để hiểu và tạo ra ngôn ngữ-thi giác thông nhất," bản in trước arXiv arXiv:2201.12086, 2022.
- [18] Y. Luo *et al.*, "Bộ chuyển đổi công tác hai cấp để chú thích hình ảnh," trong Biên bản Hội nghị về Trí tuệ nhân tạo AAAI, tập 35, số 3, 2021, trang 2286–2293.
- [19] Q. Guo, X. Qiu, P. Liu, Y. Shao, X. Xue và Z. Zhang, "Máy biến áp sao," bản in trước arXiv arXiv:1902.09113, 2019.
- [20] J. Hao, X. Wang, B. Yang, L. Wang, J. Zhang và Z. Tu, "Mô hình hóa sự tái phát cho máy biến áp," bản in trước arXiv arXiv:1904.03092, 2019.
- [21] M.-T. Luong, H. Pham và CD Manning, "Những cách tiếp cận hiệu quả đối với dịch máy thần kinh dựa trên sự chú ý," bản in trước arXiv arXiv:1508.04025, 2015.
- [22] A. Raganato, Y. Scherrer và J. Tiedemann, "Các mẫu tự chú ý của bộ mã hóa cố định trong dịch máy dựa trên bộ biến áp," bản in trước arXiv arXiv:2002.10260, 2020.
- [23] Y. Tay, D. Bahri, D. Metzler, D.-C. Juan, Z. Zhao và C. Zheng, "Synthesizer: Rethinking self-attention for transformer models," trong Hội nghị quốc tế về học máy. PMLR, 2021, tr. 10 183–10 192.
- [24] H. Ramsauer *et al.*, "Mạng Hopfield là tất cả những gì bạn cần," bản in trước arXiv arXiv:2008.02217, 2020.
- [25] W. You, S. Sun và M. Iyyer, "Sự chú ý theo kiểu Gauss được mã hóa cứng cho thần kinh dựa trên sự chú ý," bản in trước của arXiv arXiv:2005.00742, 2020.
- [26] J. Lee-Thorp, J. Ainslie, I. Eckstein và S. Ontanon, "Fnet: Trộn mã thông báo với biến đổi Fourier," bản in trước arXiv arXiv:2105.03824, 2021.
- [27] IO Tolstikhin và cộng sự, "Mlp-mixer: Kiến trúc toàn mlp cho tầm nhìn," Những tiến bộ trong hệ thống xử lý thông tin thần kinh, tập 34, trang 24 261–24 272, 2021.
- [28] R. Socher và L. Fei-Fei, "Kết nối các phương thức: Phân đoạn bán giám sát và chú thích hình ảnh bằng cách sử dụng các tập hợp văn bản không cần chỉnh," trong Hội nghị về Thị giác máy tính và Nhận dạng mẫu của Hiệp hội máy tính IEEE năm 2010. IEEE, 2010, tr. 966–973.
- [29] BZ Yao, X. Yang, L. Lin, MW Lee, và S.-C. Zhu, "I2t: Phân tích hình ảnh thành mô tả văn bản," Biên bản báo cáo của IEEE, tập 98, số 8, trang 1485–1508, 2010.
- [30] L. Wang, Z. Bai, Y. Zhang và H. Lu, "Hiển thị, nhớ lại và kể lại: Chú thích hình ảnh với cơ chế nhớ lại," trong Biên bản Hội nghị về Trí tuệ nhân tạo AAAI, tập 34, số 07, 2020, tr. 12 176–12 183.
- [31] L. Huang, W. Wang, J. Chen và X.-Y. Wei, "Chú ý đến chú ý để chú thích hình ảnh," trong Biên bản Hội nghị quốc tế IEEE về tầm nhìn máy tính, 2019, trang 4634–4643.
- [32] J.-H. Kim, J. Jun và B.-T. Zhang, "Mạng lưới chú ý song tuyến tính," bản in trước arXiv arXiv:1805.07932, 2018.
- [33] S. Sukhbaatar, E. Grave, G. Lample, H. Jegou và A. Joulin, "Tăng cường sự chú ý vào bản thân bằng trí nhớ dài dảng," bản in trước arXiv arXiv:1907.01470, 2019.
- [34] A. Gulati *et al.*, "Conformer: Bộ biến đổi tăng cường tích chập để nhận dạng giọng nói," bản in trước arXiv arXiv:2005.08100, 2020.
- [35] J. Ji *et al.*, "Cải thiện chú thích hình ảnh bằng cách tận dụng biểu diễn toàn cầu trong và giữa các lớp trong mạng máy biến áp," trong Biên bản báo cáo hội nghị AAAI về trí tuệ nhân tạo, tập 35, số 2, 2021, tr. 1655–1663.
- [36] M. Cornia, M. Stefanini, L. Baraldi và R. Cucchiara, "Bộ chuyển đổi bộ nhớ dạng lưỡi đê chú thích hình ảnh," trong Biên bản Hội nghị IEEE/CVF về Thị giác máy tính và Nhận dạng mẫu, 2020, trang 10 578–10 587.
- [37] P. Zeng, H. Zhang, J. Song và L. Gao, "Máy biến áp S2 để chú thích hình ảnh," trong Biên bản Hội nghị chung quốc tế về trí tuệ nhân tạo, tập 5, 2022.
- [38] P. Wang *et al.*, "Thông nhất các kiến trúc, nhiệm vụ và phương thức thông qua một khuôn khổ học tập trình tự sang trình tự đơn giản," bản in trước arXiv arXiv:2202.03052, 2022.
- [39] X. Hu *et al.*, "Mô rộng quy mô đào tạo trước ngôn ngữ thi giác để chú thích hình ảnh," trong Biên bản Hội nghị IEEE/CVF về Thị giác máy tính và Nhận dạng mẫu, 2022, trang 17 980–17 989.
- [40] J. Li, D. Li, C. Xiong và S. Hoi, "Blip: Khởi động quá trình đào tạo trước ngôn ngữ-hình ảnh để hiểu và tạo ra ngôn ngữ-thi giác thông nhất," bản in trước arXiv arXiv:2201.12086, 2022.
- [41] T.-Y. Lin *et al.*, "Microsoft coco: Các đối tượng chung trong ngữ cảnh," trong hội nghị châu Âu về tầm nhìn máy tính. Springer, 2014, tr. 740–755.
- [42] R. Xiong *et al.*, "Về chuẩn hóa lớp trong kiến trúc máy biến áp," trong Hội nghị quốc tế về học máy. PMLR, 2020, tr. 10 524–10 533.
- [43] SJ Rennie, E. Marcheret, Y. Mroueh, J. Ross và V. Goel, "Huấn luyện trình tự phê bình để chú thích hình ảnh," trong Biên bản Hội nghị IEEE về Thị giác máy tính và Nhận dạng mẫu, 2017, trang 7008–7024.
- [44] R. Luo, "Một biến thể tốt hơn của đào tạo trình tự phê bình," bản in trước arXiv arXiv:2003.09971, 2020.
- [45] A. Karpathy và L. Fei-Fei, "Cân chỉnh ngữ nghĩa thi giác sâu để tạo ra mô tả hình ảnh," trong Biên bản báo cáo hội nghị IEEE về thi giác máy tính và Nhận dạng mẫu, 2015, trang 3128–3137.
- [46] H. Agrawal và cộng sự, "Nocaps: Chú thích đối tượng mới ở quy mô lớn," trong Biên bản Hội nghị quốc tế IEEE/CVF về Tầm nhìn máy tính, 2019, trang 8948–8957.
- [47] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li và L. Fei-Fei, "Imagenet: Cơ sở dữ liệu hình ảnh phân cấp quy mô lớn," trong hội nghị IEEE năm 2009 về thi giác máy tính và Nhận dạng mẫu. Ieee, 2009, tr. 248–255.
- [48] JC Hu, R. Cavigchioli và A. Capotondi, "Khám phá nút thắt về độ dài chuỗi trong bộ chuyên đổi để chú thích hình ảnh," 2022.
- [49] J. Hu, R. Cavigchioli và A. Capotondi, "Yêu cầu làm rõ về mã thông báo kết thúc chuỗi trong quá trình đào tạo chuỗi tự phê bình," Ghi chú bài giảng về Khoa học máy tính (bao gồm các tiêu mục Ghi chú bài giảng về Trí tuệ nhân tạo và Ghi chú bài giảng về Tin sinh học), tập 14233 LNCS, trang 39 – 50, 2023.

- [50] T. Yao, Y. Pan, Y. Li, and T. Mei, "Exploring visual relationship for image captioning," in *Proceedings of the European conference on computer vision (ECCV)*, 2018, pp. 684–699.
- [51] X. Yang, K. Tang, H. Zhang, and J. Cai, "Auto-encoding scene graphs for image captioning," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2019, pp. 10 685–10 694.
- [52] X. Zhang *et al.*, "Rstnet: Captioning with adaptive attention on visual and non-visual words," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2021, pp. 15 465–15 474.
- [53] L. Liu *et al.*, "On the variance of the adaptive learning rate and beyond," *arXiv preprint arXiv:1908.03265*, 2019.
- [54] P. Anderson, B. Fernando, M. Johnson, and S. Gould, "Guided open vocabulary image captioning with constrained beam search," *arXiv preprint arXiv:1612.00576*, 2016.
- [55] S. Changpinyo, P. Sharma, N. Ding, and R. Soricut, "Conceptual 12m: Pushing web-scale image-text pre-training to recognize long-tail visual concepts," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2021, pp. 3558–3568.
- [56] P. Zhang *et al.*, "Vinvl: Revisiting visual representations in vision-language models," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2021, pp. 5579–5588.
- [50] T. Yao, Y. Pan, Y. Li và T. Mei, "Khám phá mối quan hệ trực quan để chú thích hình ảnh", trong Biên bản hội nghị châu Âu về thị giác máy tính (ECCV), 2018, trang 684–699.
- [51] X. Yang, K. Tang, H. Zhang và J. Cai, "Đồ thị cảnh tự động mã hóa để chú thích hình ảnh," trong Biên bản báo cáo Hội nghị IEEE/CVF về Thị giác máy tính và Nhận dạng mẫu, 2019, trang 10 685–10 694.
- [52] X. Zhang và cộng sự, "Rstnet: Chú thích với sự chú ý thích ứng trên các từ ngữ trực quan và không trực quan," trong Biên bản báo cáo hội nghị IEEE/CVF về Thị giác máy tính và Nhận dạng mẫu, 2021, trang 15 465–15 474.
- [53] L. Liu và cộng sự, "Về sự thay đổi của tốc độ học tập thích ứng và hơn thế nữa," bản in trước arXiv arXiv:1908.03265, 2019.
- [54] P. Anderson, B. Fernando, M. Johnson và S. Gould, "Chú thích hình ảnh từ vựng mở có hướng dẫn với tìm kiếm chùm tia bị hạn chế", bản in trước arXiv arXiv:1612.00576, 2016.
- [55] S. Changpinyo, P. Sharma, N. Ding và R. Soricut, "Conceptual 12m: Đầy mạnh quá trình đào tạo trước hình ảnh-văn bản ở quy mô web để nhận dạng các khái niệm trực quan đầu dài", trong Biên bản báo cáo Hội nghị IEEE/CVF về Thị giác máy tính và Nhận dạng mẫu, 2021, trang 3558–3568.
- [56] P. Zhang và cộng sự, "Vinvl: Xem xét lại các biểu diễn trực quan trong các mô hình ngôn ngữ thị giác," trong Biên bản báo cáo hội nghị IEEE/CVF về Thị giác máy tính và Nhận dạng mẫu, 2021, trang 5579–5588.