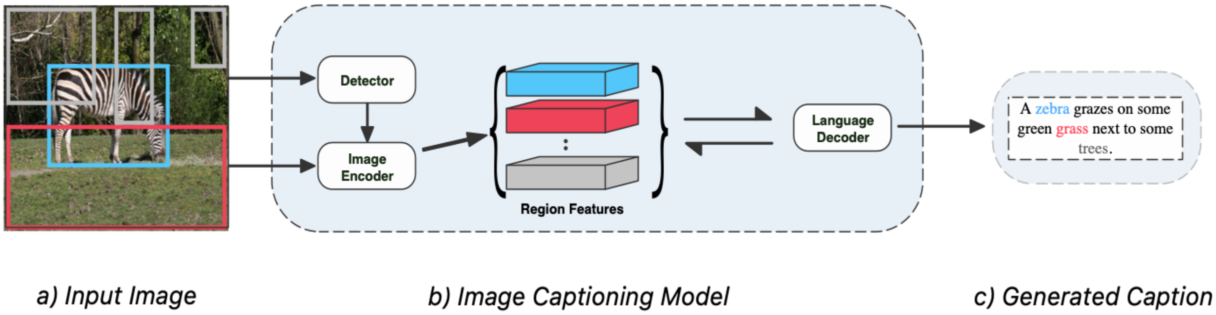
Image Understanding by Captioning with Differentiable Architecture Search

Ramtin Hosseini
rhossein@ucsd.edu
University of California San Diego
San Diego, CA, USA

Pengtao Xie
p1xie@eng.ucsd.edu
University of California San Diego
San Diego, CA, USA



a) Input Image

b) Image Captioning Model

c) Generated Caption

Figure 1: Overview of an *Image Captioning* task. Unlike most of the existing methods that utilize fixed human-designed architectures as their image encoders and language decoders, Image Understanding by Captioning (IUC) searches for the most suitable task-specific encoder-decoder architectures in a multi-level optimization (MLO) framework.

ABSTRACT

In deep learning applications, image understanding is a crucial task, where several techniques such as image captioning and visual question answering have been widely studied to improve and evaluate the performances of deep neural networks (DNN) in this area. In image captioning, models have encoder-decoder architectures, where the encoders take the input images, produce embeddings, and feed them into the decoders to generate textual descriptions. Designing a proper image captioning encoder-decoder architecture manually is a difficult challenge due to the complexity of recognizing the critical objects of the input images and their relationships to generate caption descriptions. To address this issue, we propose a three-level optimization method that employs differentiable architecture search strategies to seek the most suitable architecture for image captioning automatically. Our optimization framework involves three stages, which are performed end-to-end. In the first stage, an image captioning model learns and updates the weights of its encoder and decoder to create image captions. At the next stage, the trained encoder-decoder generates a pseudo image captioning dataset from unlabeled images, and the predictive model trains on the generated dataset to update its weights. Finally, the trained model validates its performance on the validation set and updates the encoder-decoder architecture by minimizing the validation loss. Experiments and studies on the COCO image captions datasets demonstrate that our method performs significantly better than the baselines and can achieve state-of-the-art results in image understanding tasks.

MM '22, October 10–14, 2022, Lisboa, Portugal

ACM ISBN 978-1-4503-9203-7/22/10.
https://doi.org/10.1145/3503161.3548150

CCS CONCEPTS

• **Networks** → **Network architectures**; • **Computing methodologies** → **Machine learning algorithms**; **Neural networks**.

KEYWORDS

Image Captioning; Neural Architecture Search; Multilevel Optimization; Deep Learning; Visual reasoning and logical representation; Computer Vision + Natural Language Processing (NLP)

ACM Reference Format:
Ramtin Hosseini and Pengtao Xie. 2022. Image Understanding by Captioning with Differentiable Architecture Search. In *Proceedings of the 30th ACM International Conference on Multimedia (MM '22), Oct. 10–14, 2022, Lisboa, Portugal.* ACM, New York, NY, USA, 9 pages. https://doi.org/10.1145/3503161.3548150

1 INTRODUCTION

The rapid advancement of deep learning techniques is assisting with resolving social difficulties in various fields. One of these fields that has recently attracted the researchers' attention is image understanding (e.g., image captioning). *Image Captioning* is a multi-model task that evaluates the computer's ability to understand images by generating the language descriptions of the input images, as shown in Figure 1. Solving this visual-language problem can be challenging, considering the complexity of understanding the relationships of recognized critical objects in the images. Since the architecture design of deep neural networks plays a critical role in the performance enhancement of the model, researchers have proposed a significant number of techniques in order to enhance the performance of their models by designing proper architectures for the encoder and the decoder modules for various tasks. However, obtaining high-performance human-designed architectures for each dataset with different distributions is exhausting and time-consuming.

Recently, neural architecture search (NAS) has achieved remarkable progress in obtaining the optimal architectures automatically, which helps attain better performances in computer vision and natural language processing applications. Nevertheless, most researchers in this area have focused on applying NAS methods to language modeling [41], image classification[9, 11, 26, 34, 42, 43], and adversarial training[12], while image captioning study with NAS is still largely underexplored. Several works have been investigated in employing NAS methods on image captioning, such as [41], where they focus only on searching for the architecture of the decoder module (i.e., language generation module) by using Reinforcement Learning [41].

In this paper, we propose a three-stage optimization problem, called Image Understanding by Captioning (IUC), that applies differentiable architecture search [22] on image captioning tasks. Unlike the previous related works [41] that apply Reinforcement Learning based (RL-based) NAS methods only on decoder modules, we utilize differentiable architecture search based (DARTS-based) approaches on both encoder and decoder modules to improve our model's performance and, also, study the effectiveness and importance of each module. Extensive experiments on COCO datasets [20] explicate that our proposed methods outperform the existing strategies and can achieve state-of-the-art performances in image captioning.

The main contributions of this paper include:

- We propose a novel three-level optimization framework for image captioning that utilizes differentiable architecture search to obtain the optimal encoder-decoder architecture automatically. Additionally, our proposed methods can be applied on top of any differentiable NAS methods for further improvements.
- We investigate the effectiveness and impact of the architecture design in the encoder modules compared to the decoder modules. Furthermore, we demonstrate that the design of the image encoder architecture has a higher effect on the image captioning performances than the design of the language decoder.
- Extensive experiments and studies of both qualitative and quantitative results on COCO image captions datasets illustrate that our Image Understanding by Captioning (IUC) method is superior to the existing methods and achieves state-of-the-art performance in image captioning tasks across various metrics.

2 RELATED WORKS

2.1 Image Captioning

Image Captioning is a multi-modal task that generates textual descriptions of input images. In order to perform this vision-language process, we need an encoder-decoder framework, where the encoders take the input images and create embeddings to feed them into the decoders to generate captions. Various techniques have been introduced in the past few years to enhance image captioning. Early works [23] in this area mostly use CNN as the image encoder, and LSTM [32] and RNN [38] as the language decoder to generate the captions of the input images. Later on, various attempts [6] showed performance improvements by applying attention mechanisms for more information exchange between the encoder and decoder modules. In several recent works, significant progress has been made with transformers [7] architectures. On the other hand, various approaches have been made to enhance the object detection task part of the image captioning, such as employing grid feature, region feature, and relation-aware visual feature. In the most recent works [18], researchers exhibit that vision-language pre-training on large image-text datasets can improve image captioning performances significantly. Moreover, several other recent works have been studying the importance of the image-captioning encoder-decoder architecture [3] by investigating different encoder-decoder architectures' performances versus their model sizes. Despite all the progress that has been made in image captioning tasks over the past decade, most of the existing image captioning models suffer from the design of their encoder and decoder architectures, which are fixed human-designed. Recently, AutoCaption [41] has proposed applying reinforcement learning-based neural architecture search (NAS) to image captioning tasks in order to design a better language decoder on the X-LAN [24]. Since reinforcement learning-based neural architecture search methods are mainly expensive for computer vision tasks (e.g., image classification), the existing image captioning methods that utilize NAS are inefficient in searching for the image encoder architecture. To address this problem, we propose a novel method that uses differentiable architecture search techniques to obtain the optimal task-specific image encoder and language decoder architectures for image captioning tasks.

2.2 Neural Architecture Search (NAS)

Recently, a wide variety of NAS methods have been proposed and achieved considerable success in automatically identifying highly-performing architectures of neural networks to reduce reliance on human experts. Thus, These NAS approaches have attracted plenty of attention in deep learning applications of computer vision and natural language processing tasks, such as image classification, object detection, and language modeling, to automatically design suitable neural network architectures. Early NAS approaches [26, 42, 43] are mainly based on reinforcement learning (RL) which uses a policy network to generate architectures and evaluate these architectures on the validation set. The validation loss is used as a reward to optimize the policy network and train it to produce high-quality architectures. While RL-based approaches achieve the first wave of success in NAS research, they are computationally costly since evaluating the architectures requires a heavy-duty training process. This limitation renders RL-based approaches not applicable for most users with insufficient computational resources. To address this issue, differentiable search methods [22] have been proposed, which parameterize architectures as differentiable functions and perform a search using efficient gradient-based methods. In these methods, the search space of architectures is composed of a large set of building blocks where the output of each block is multiplied with a smooth variable indicating how important this block is. Under such a formulation, search solves a mathematical optimization problem defined on the important variables where the objective is to find an optimal set of variables that yield the lowest validation loss. This optimization problem can be solved by gradient-based methods. Differentiable NAS research is initiated by DARTS [22] and further improved by subsequent works such as P-DARTS [34], PC-DARTS [33], DATA [5], etc. P-DARTS [34] grows the depth of architectures progressively in the search process. PC-DARTS [33] samples sub-architectures from a super network to reduce redundancy during the search. DATA [5] proposes using Gumbel-Softmax to change the weight vectors of the operations to one-hot or binary code, which reduces the gap between the architectures in the searching and validating stages. Most recent works [10] additionally apply topology search to gradient-based methods due to the high effectiveness of topology on neural networks. Besides RL-based approaches and differentiable NAS, another paradigm of NAS methods [21, 28] are based on the evolutionary algorithm. In these methods, architectures are formulated as individuals in a population. High-quality architectures produce offspring to replace low-quality architectures, where the quality is measured using fitness scores. Similar to RL-based approaches, these methods also require considerable computing resources.

3 METHODS

In this section, we propose a novel method called Image Understanding by Captioning (IUC), where we apply differentiable architecture search techniques to obtain the optimal architectures for the encoder-decoder model. Inspired by [22], the architecture cells are directed acyclic graphs (DAG) with N nodes (i.e., latent representations) and directed edges, representing the operation between the corresponding nodes. Our three-level optimization framework contains an encoder-decoder image captioning model with searchable architecture and a predictive image captioning (IC) model with fixed human-designed architecture. The searchable encoder-decoder model learns to take an image and generate text descriptions of the given image. Then, using the learned encoder-decoder model, we generate a pseudo image captioning dataset from the unlabeled dataset, and we train our predictive model on the new pseudo-IC dataset. Lastly, the trained predictive model validates its performance on the validation set of the IC dataset and minimizes its validation loss. To independently investigate the effectiveness of the image encoder module and language decoder module, we perform image encoder and language decoder architecture searches individually. Figure 2 illustrates our three-level optimization framework (IUC), where the solid arrows represent that the predictions are made and training/validation losses are determined; and the dotted arrows denote that the gradient updates of network weights and architecture variables are determined and weights/architecture are updated.

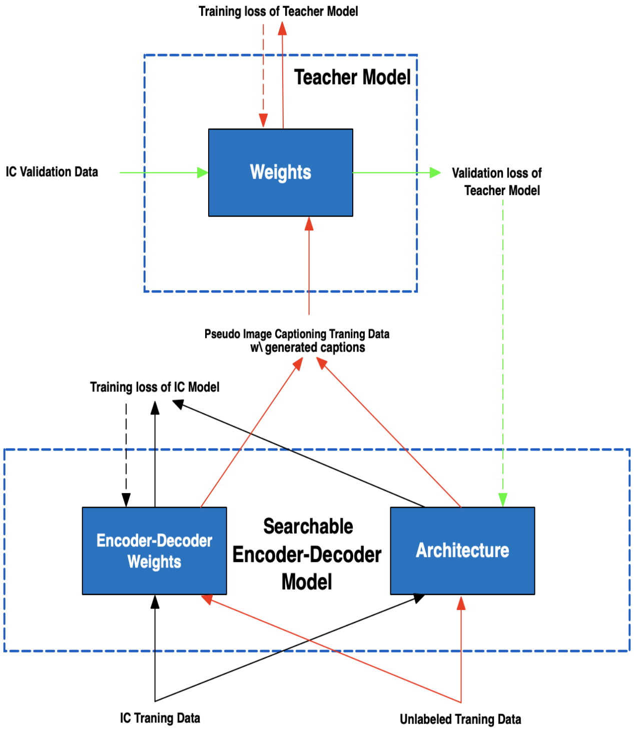

Training loss of Teacher Model
Teacher Model
IC Validation Data
Weights
Validation loss of Teacher Model
Pseudo Image Captioning Traning Data w\ generated captions
Training loss of IC Model
Encoder-Decoder Weights
Searchable Encoder-Decoder Model
Architecture
IC Traning Data
Unlabeled Traning Data

Figure 2: Overview of our Image Understanding by Captioning (IUC) optimization framework. Black, red, and green arrows represent stage 1, stage 2, and stage 3 in our framework, respectively.

3.1 Proposed Framework (IUC)

In our framework, there are three learning stages. In the **first stage**, a model learns to create image captions. The model has an encoder-decoder architecture. The encoder takes an input image and produces an embedding. The embedding is fed into the decoder, which decodes a textual description. We use the image captioning datasets to train the encoder and decoder by solving the following problem:

$$E^*(A), F^*(A) = \underset{E,F}{\operatorname{argmin}}\ L(E, A, F, D^{(tr)}) \qquad (1)$$

where F and E denote the network weights of the decoder and the encoder, respectively, and A represents the architectures of the encoder and the decoder. $D^{(tr)}$ is an image captioning dataset.

In the **second stage**, we use the trained encoder $E^*(A)$ and decoder $F^*(A)$ with searchable architectures from the first stage to generate a pseudo image captioning dataset using unlabeled images

Algorithm 1: Optimization algorithm for image understanding by captioning

```
while not converged do
    1. Update encoder weights:
        $E \leftarrow E - \eta_e \nabla_E L(E, A, F, D^{(tr)})$
    2. Update decoder weights:
        $F \leftarrow F - \eta_f \nabla_F L(E, A, F, D^{(tr)})$
    3. Update predictive model weights:
        $W \leftarrow W - \eta_w \nabla_W L(W, U, E', F')$
    4. Update the encoder-decoder architecture:
        $A \leftarrow A - \eta_a \nabla_A L(W', D^{(val)}))$
end
```

U. Then, using this new dataset, we train the network weights W of the predictive model with a fixed architecture by minimizing the following training loss:

$$W^*(E^*(A), F^*(A)) = \underset{W}{\operatorname{argmin}} \ L(W, U, E^*(A), F^*(A)) \tag{2}$$

Finally, we evaluate $W^*(E^*(A), F^*(A))$ in the **third stage** on the image captioning validation set and update A by minimizing the validation loss:

$$\min_{A} \ L(W^*(E^*(A), F^*(A)), D^{(val)}) \tag{3}$$

The predictive model in our framework assists the initial image captioning model to obtain the optimal architecture by testing its caption generating performance. Putting these pieces together, we get the following optimization problem:

$$\begin{aligned} &\min_{A} \ L(W^*(E^*(A), F^*(A)), D^{(val)}) \\ &s.t. \ \ W^*(E^*(A), F^*(A)) = \underset{W}{\operatorname{argmin}} \ L(W, U, E^*(A), F^*(A)) \\ &\qquad E^*(A), F^*(A) = \underset{E,F}{\operatorname{argmin}} \ L(E, A, F, D^{(tr)}) \end{aligned} \tag{4}$$

The architecture searches for the image encoder and the language decoder modules are similar to Conventional Cell Search and Recurrent Cell Search as proposed in DARTS [22], respectively. The encoder during the architecture search contains 8 optimal cells, and the decoder is a single cell. Our proposed IUC framework is orthogonal to the various differentiable NAS methods, and it can be employed on any DARTS-based method, including DARTS [22], P-DARTS [34], PC-DARTS [33], and DATA [5].

Image Encoder Architecture Search. In order to obtain the optimal image encoder architecture for image captioning on a particular dataset, we search for convolutional cells similar to DARTS [22] to optimize the encoder module architecture A using Eq.4. Inspired by [22], the encoder module is a convolution network built by stacking the learned cells together. The search spaces for the encoder architectures include (dilated) separable convolutions with sizes of 3 × 3 and 5 × 5, max pooling with the size of 3 × 3, average pooling with the size of 3 × 3, identity, and zero.

Language Decoder Architecture Search. To design the language decoder module, we perform the architecture searching for recurrent cells analogous to DARTS [22]. In this case, The architecture A in Eq.4 represents the decoder architecture, where the learned cells are recursively connected in order to build the recurrent network of the language decoder. Our defined primitive operations in the search space for the recurrent cell of the language decoder include linear transformations followed by activation functions, the identity mapping, and the zero operation. The activation functions are chosen from one of the following: *relu*, *tanh*, *sigmoid*, *elu*, *celu*, or *gelu*.

Notation	Meaning
E	Encoder weights
F	Decoder weights
W	Network weights of predictive model
A	Encoder-Decoder architecture
$D^{(tr)}$	Image captioning training set
$D^{(val)}$	Image captioning validation set
U	Unlabeled image dataset
η_e	Learning rate of E
η_f	Learning rate of F
η_w	Learning rate of W
η_a	Learning rate of A

Table 1: Notations in Image Understanding by Captioning (IUC).

3.2 Optimization Algorithm

In this section, we develop an optimization algorithm to solve the problem in Eq.(4) with our defined notations from Table 1. We approximate $E^*(A)$ and $F^*(A)$ using one-step gradient descent w.r.t $L(E, A, F, D^{(tr)})$:

$$E^*(A) \approx E' = E - \eta_e \nabla_E L(E, A, F, D^{(tr)}) \tag{5}$$

$$F^*(A) \approx F' = F - \eta_f \nabla_F L(E, A, F, D^{(tr)}) \tag{6}$$

We plug E' and F' into $L(W, U, E^*(A), F^*(A))$ and get an approximated objective. We approximate $W^*(E^*(A), F^*(A))$ using one-step gradient descent w.r.t the approximated objective:

$$W^*(E^*(A), F^*(A)) \approx W' = W - \eta_w \nabla_W L(W, U, E', F') \tag{7}$$

We plug W' into $L(W^*(E^*(A), F^*(A)), D^{(val)})$ and get an approximated objective. Thus, we update A using gradient descent:

$$A \leftarrow A - \eta_a \nabla_A L(W', D^{(val)}), \tag{8}$$

where by applying chain rule to the approximate architecture gradient Eq. 8, we get:

$$\begin{gathered} \nabla_A L(W', D^{(val)}) = (\frac{\partial E'}{\partial A}\frac{\partial W'}{\partial E'} + \frac{\partial F'}{\partial A}\frac{\partial W'}{\partial F'})\nabla_{W'} L(W', D^{(val)}) = \\ (\eta_e \eta_w \nabla^2_{A,E} L(E, A, F, D^{(tr)}) \nabla^2_{E',W} L(W, U, E', F') + \eta_f \eta_w \nabla^2_{A,F} \\ L(E, A, F, D^{(tr)})) \nabla^2_{F',W} L(W, U, E', F')) \nabla_{W'} L(W', D^{(val)}) \end{gathered} \tag{9}$$

The overall algorithm of IUC is shown in Algorithm 1.

Table 2: Comparison of our methods and the state-of-the-art image captioning models on the COCO "Karpathy" test split (single-model). Methods with † are using NAS methods.

	Cross-Entropy Loss						Encoder-Decoder
	BLEU-1	BLEU-4	METEOR	ROUGE	CIDEr	SPICE	Architecture
LSTM [32]	-	29.6	25.2	52.6	94.0	-	manual
SCST [30]	-	30.0	25.9	53.4	99.4	-	manual
LSTM-A [37]	75.4	35.2	26.9	55.8	108.8	20.0	manual
RFNet [14]	76.4	35.8	27.4	56.5	112.5	20.5	manual
Up-Down [2]	77.2	36.2	27.0	56.4	113.5	20.3	manual
GCN-LSTM [36]	77.3	36.8	27.9	57.0	116.3	20.9	manual
LBPF [27]	77.8	37.4	28.1	57.5	116.4	21.2	manual
SGAE [35]	77.6	36.9	27.7	57.2	116.7	20.9	manual
AoANet [13]	77.4	37.2	28.4	57.5	119.8	21.3	manual
X-LAN [24]	78.0	38.2	28.8	58.0	122.0	21.9	manual
X-Transformer[24]	77.3	37.0	28.7	57.5	120.0	21.8	manual
OSCAR$_L$ [18]	-	37.4	30.7	-	127.8	23.5	manual
OSCAR+$_L$ w/ VINVL [40]	-	38.5	30.4	-	130.8	23.4	manual
AutoCaption [41]†	79.4	39.2	29.0	58.6	125.2	22.4	RL
IUC-D (ours) †	79.6	39.5	30.8	58.9	130.6	**23.8**	gradient-based
IUC-E (ours) †	**79.9**	**40.0**	**30.9**	**59.3**	**131.1**	23.7	gradient-based

	CIDEr Score Optimization						Encoder-Decoder
	BLEU-1	BLEU-4	METEOR	ROUGE	CIDEr	SPICE	Architecture
LSTM [32]	-	31.9	25.5	54.3	106.3	-	manual
SCST [30]	-	34.2	26.7	55.7	114.0	-	manual
LSTM-A [37]	78.6	35.5	27.3	56.8	118.3	20.8	manual
RFNet [14]	79.1	36.5	27.7	57.3	121.9	21.2	manual
Up-Down [2]	79.8	36.3	27.7	56.9	120.1	21.4	manual
GCN-LSTM [36]	80.5	38.2	28.5	58.3	127.6	22.0	manual
LBPF [27]	80.5	38.3	28.5	58.4	127.6	22.0	manual
SGAE [35]	80.8	38.4	28.4	58.6	127.8	22.1	manual
AoANet [13]	80.2	38.9	29.2	58.8	129.8	22.4	manual
X-LAN [24]	80.8	39.5	29.5	59.2	132.0	23.4	manual
X-Transformer[24]	80.9	39.7	29.5	59.1	132.8	23.4	manual
Meshed-Memory Transformer [7]	80.8	39.1	29.2	58.6	131.2	22.6	manual
X-Transformer+PPO [39]	81.1	39.7	29.6	59.2	133.3	23.4	manual
OSCAR$_L$ [18]	-	41.7	30.6	-	140.0	24.5	manual
OSCAR+$_L$ w/ VINVL [40]	-	41.0	31.1	-	140.9	25.2	manual
AutoCaption [41] †	81.5	40.2	29.9	59.5	135.8	23.8	RL
IUC-D (ours) †	81.8	40.9	31.0	59.5	140.6	25.3	gradient-based
IUC-E (ours) †	**82.3**	**42.1**	**31.4**	**60.1**	**141.9**	**25.8**	gradient-based

4 EXPERIMENTS

In this section, we apply our proposed IUC methods to perform image encoder and language decoder architecture searches. Each experiment consists of two steps: architecture search and architecture evaluation. The optimal cell is obtained in the search process, and it will be evaluated in the evaluation stage based on the formed large network from the optimal cell. The large network will be retrained from scratch for the architecture evaluation.

4.1 Datasets

We perform experiments on the COCO captions dataset [20] to evaluate and compare our proposed methods. The COCO captions dataset contains 82, 783 and 40, 504 images in the training and validation sets, respectively. We conduct thorough experiments by analyzing our models on the offline and the online evaluations. Each image in the dataset holds five captions, which were annotated by humans. During the offline evaluation, we utilize the 'Karpathy' splits setting [15], which has 113, 287 images and 5000 images in the training set and test set, respectively. At the second stage of the architecture search, we use the 123K unlabeled images of the COCO dataset, which has a similar class distribution as the labeled images, and we use our trained encoder-decoder model from the first stage to generate a textual description of the images. Then the pre-trained predictive IC model will be trained based on the generated pseudo-dataset. Finally, we update the architecture of the encoder-decoder by minimizing the validation loss of the predictive model on the validation set.

4.2 Experimental Settings

Recall that our proposed framework IUC is a comprehensive differentiable method, which can be applied with any differentiable architecture search approach. With that being said, we employ DARTS-2nd [22] in the conducted experiments that are exhibited in Table 2, 4, and 5, and DARTS [22], PC-DARTS [33], and DATA [5] in Table 6 of the Ablation 4.4, which is shown in Table 6. We introduce two variants of our IUC method: 1) IUC with image encoder architecture search (IUC-E); and 2) IUC with language decoder architecture search (IUC-D). Moreover, we simultaneously apply IUC architecture searching on both the image encoder and the language decoder in Ablation 4.4. Note that the common operations in differentiable architecture searches usually do not contain some practical operations such as: attention, top-down, or RoI pooling. To address this issue, we modify our image encoder and language decoder inspired by some SoTA methods such as Faster R-CNN [29] and X-LAN [24] techniques.

Architecture Search Details. During the search stage, we use Faster R-CNN [29] and X-LAN [24] as our image encoder and sentence decoder of the image captioning model, respectively. In IUC-E, we switch their human-designed architecture network with 8 convolutional cells (each holding 7 nodes), while during the IUC-D search we replace their LSTMs with our recurrent cell, which consists of 12 nodes. In IUC-D, ResNeXt-152 is adopted as the CNN in the Faster R-CNN. In addition, the initial convolutional architectures of the image encoder in IUC-E are pre-trained on ImageNet [8], and Visual Genome [17] datasets prior to the search for improving the object feature extractions. Following the settings in DARTS [22], we use SGD for the IUC-E architecture searching with a batch size of 128, an initial learning rate of 0.025, weight decay of 3e-4, and a momentum of 0.9 for 50 epochs. More detailed hyperparameter settings are exhibited in the Appendix. Due to high-performance achievements of *OSCAR* [18], we use pre-trained $OSCAR_L$ [18] as our predictive model to help us obtain the optimal encoder-decoder architecture. At the second stage of architecture search, the trained encoder-decoder from the first stage generates a textual description of the input image. Then, the predictive model learns from the generated pseudo dataset and validates its performance on the validation set to update the architecture of our encoder-decoder. In this paper, we constructed our experiments with similar settings as $OSCAR_L$ [18] for a fair comparison.

Architecture Evaluation Details. For a fair comparison, we adopt X-LAN [24] method as our language decoder and Faster R-CNN [29] as the image decoder. In IUC-E, we replace the convolution neural network architecture of Faster R-CNN [29] with our obtained image encoder architecture, designed by stacking 14 searched optimal convolution cells. Later, we pre-train the model on ImageNet [8] and Visual Genome [17] to extract the object tags and image region features. Similarly, in IUC-D, we use Faster R-CNN [29] with ResNeXt-152 pre-trained on ImageNet and Visual Genome dataset datasets, and we change the LSTMs of the X-LAN with our optimal architecture that was obtained during the architecture search. Then, we train our constructed large network with cross-entropy loss for 100 epochs and 10000 warmup steps. Next, we choose the model that achieves the highest CIDEr score, and we CIDEr score optimization with the learning rate of 0.00001 for another 100 epochs. During the validation, we utilize the beam search with the beam size of 3 and our models are trained with Adam optimizer [16].

Metrics. We adopt the official evaluation metrics - including BLEU-N[25], METEOR [4], ROUGE[19], CIDEr [31], and SPICE[1] - to analyze and compare our proposed methods' performances with the other existing approaches in image captioning tasks.

4.3 Results

We evaluate the image captioning results of our proposed methods on the COCO "Karpathy" test split [15] to compare with the recent proposals in this area, which have achieved noteworthy performances. Our primary baselines include: LSTM [32] and LSTM-A[37], which are non-attention based; SCST[30] that proposes employing attention over the grid of features; RFNet[14] merges CNN features by adopting recurrent fusion networks; Up-Down[2] uses attention over regions; GCN-LSTM[36] uses visual relations between image regions; SGAE[35] utilizes auto-encoding scene graphs for sentence generation; AoANet[13] applies attention on attention and LSTM as the image encoder and language decoder, respectively; Meshed-Memory Transformer[7] constructs mesh-like transformer connectivity between the encoder and the decoder; X-LAN[24] plugs unified attention blocks, called X-Linear attention blocks, into the encoder-decoder architecture, and further uses such blocks in the Transformer-based encoder-decoder architecture, which is called X-Transformer[24]; OSCAR[18] adopts object tags as anchor points to enhance the learning of the image-text semantic alignments; OSCAR+ with VINVL[40] shows that visual features are crucial in image understanding tasks by improving object detection model of OSCAR+; X-Transformer+PPO[39] applies proximal policy optimization to X-Transformer; and AutoCaption [41] that applies neural architecture search on the language decoder with a similar structure as X-LAN.

Table 2 reports the performance comparisons of our proposed methods (IUC-D and IUC-E) and the state-of-the-art models on the offline COCO "Karpathy" test split for both cross-entropy loss and CIDEr score optimization. It is shown that our method IUC-E outperforms the baselines and elevates the state-of-the-art in most of the metrics, while IUC-D frequently exhibits the second-best performances among the other methods. IUC-D performs slightly better than IUC-E in SPICE. Our proposed IUC-E model can achieve 141.9 on CIDEr using CIDEr score optimization, which indicates an improvement of 1 CIDEr point compared to OSCAR+ with VINVL, and 6.1 points improvement in comparison to AutoCaption. This performance enhancement verifies the critical advantage of employing architecture search to design the encoder and the decoder architectures. Additionally, the better performances of IUC-E compared to IUC-D in the evaluation results demonstrate the urgency of the architecture design of the image encoder, which has not been investigated relatively. Finally, we ensemble our IUC-E and IUC-D

Table 3: Comparison of our methods and the state-of-the-art image captioning models on the COCO “Karpathy” test split with multiple models (Ensemble/Fusion). Methods with † are using NAS methods.

	Cross-Entropy Loss						Encoder-Decoder
	BLEU-1	BLEU-4	METEOR	ROUGE	CIDEr	SPICE	Architecture
SCST$^{\Sigma}$ [30]	-	32.8	26.7	55.1	106.5	-	manual
RFNet$^{\Sigma}$ [14]	77.4	37.0	27.9	57.3	116.3	20.8	manual
GCN-LSTM$^{\Sigma}$ [36]	77.4	37.1	28.1	57.2	117.1	21.1	manual
SGAE$^{\Sigma}$ [35]	-	-	-	-	-	-	manual
AoANet$^{\Sigma}$ [13]	78.7	38.1	28.5	58.2	122.7	21.7	manual
X-LAN$^{\Sigma}$ [24]	78.8	39.1	29.1	58.5	124.5	22.2	manual
X-Transformer$^{\Sigma}$[24]	77.8	37.7	29.0	58.0	122.1	21.9	manual
AutoCaption$^{\Sigma}$ [41] †	79.8	40.3	29.6	59.2	128.5	22.8	RL
IUC (ours) †	**80.0**	**40.6**	**31.2**	**59.4**	**132.8**	**23.9**	gradient-based

	CIDEr Score Optimization						Encoder-Decoder
	BLEU-1	BLEU-4	METEOR	ROUGE	CIDEr	SPICE	Architecture
SCST$^{\Sigma}$ [30]	-	35.4	27.1	56.6	117.5	-	manual
RFNet$^{\Sigma}$ [14]	80.4	37.9	28.3	58.3	125.7	21.7	manual
GCN-LSTM$^{\Sigma}$ [36]	80.9	38.3	28.6	58.5	128.7	22.1	manual
SGAE [35]	81.0	39.0	28.4	58.9	129.1	22.2	manual
AoANet$^{\Sigma}$ [13]	81.6	40.2	29.3	59.4	132.0	22.8	manual
X-LAN$^{\Sigma}$ [24]	81.6	40.3	29.8	59.6	133.7	23.6	manual
X-Transformer$^{\Sigma}$[24]	81.7	40.7	29.9	59.7	135.3	23.8	manual
AutoCaption$^{\Sigma}$ [41] †	82.9	42.1	30.4	60.4	139.5	24.3	RL
IUC (ours) †	**82.4**	**42.5**	**31.8**	**60.4**	**142.7**	**25.9**	gradient-based

models, called IUC, for further improvement. In Table 3, we evaluate and compare the performance of IUC and the existing works by utilizing ensemble models. Our extensive experiments show that IUC can outperform the existing image captioning models in single and ensemble model settings with both Cross-Entropy loss optimization and CIDEr Score optimization.

Figure 3 showcases different exemplar of generated captions by our IUC and our baseline (AutoCaption$^{\Sigma}$ [41]) along with the human-annotated ground truth (GT) captions. As it is shown, our model generates more accurate, clear, and detailed textual descriptions in challenging image cases, since our encoder-decoder architecture in IUC is task-specific designed, while AutoCaption [41] aims only to design the language decoder using NAS. This implies that the architecture design of the image encoder has more impact on the model’s performance than the language decoder’s architecture design.

4.4 Ablation Studies

Ablation 1. We are interested in verifying the critical advantage of employing architecture search in image captioning tasks. In this study, we employ architecture search and random sampling for the architecture designs of the image encoder and the language decoder.

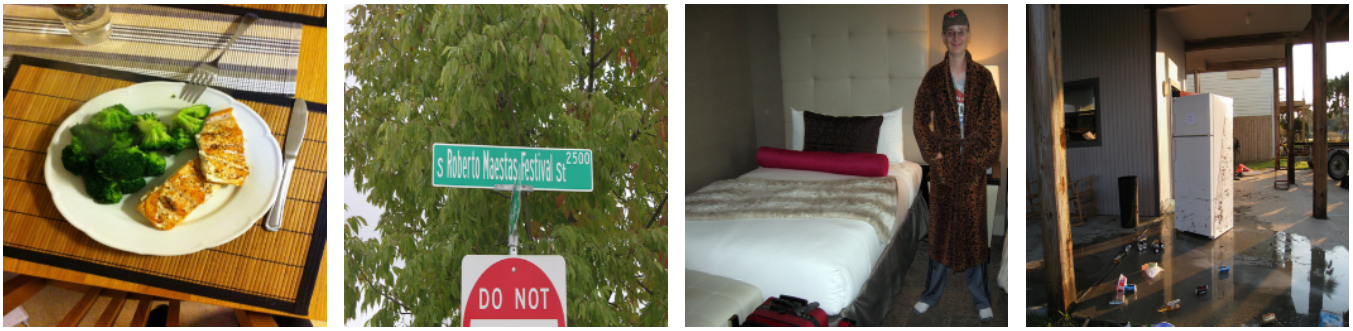
Similar to Section 4.2; first, we perform an architecture search or random sampling to obtain the optimal cells, then we construct the large network by using the optimal cells, and finally, we train and evaluate the large network. To get a deeper understanding of the impact that each module’s architecture (i.e., image encoder and language decoder) has on the image captioning performance, we do not use the OSCAR pre-training during this study - unlike the experiments in Table 2 - to reduce the constraints and dependencies of our investigation. Similar to DARTS [22], image encoder and language decoder architectures are searched or randomly sampled from convolutional or recurrent cells, respectively. Table 4 shows that architecture search can significantly enhance the image captioning models’ performance, and the architecture design of the image encoder is more crucial for achieving higher performance than the architecture design of the language decoder.

Table 4: Comparison of searched (S) and randomly sampled (R) encoder and decoder architectures on COCO “Karpathy” test split (single-model with Cross-Entropy Loss).

Encoder	Decoder	BLEU-4	METEOR	CIDEr	SPICE
R	R	37.7	28.0	117.9	21.7
R	S	37.9	28.7	122.1	21.9
S	R	**38.3**	**29.4**	**127.3**	**22.6**

Ablation 2. In this setting, we investigate how the IUC-E model’s performance varies as the tradeoff parameters λ and γ change in Eq.(10).

$$\begin{aligned} \min_{A} \quad & \lambda L(W^*(E^*(A), F^*(A)), D^{(val)}) + \gamma L(E^*(A), F^*(A), D^{(val)}) \\ s.t. \quad & W^*(E^*(A), F^*(A)) = \underset{W}{\operatorname{argmin}} \; L(W, U, E^*(A), F^*(A)) \qquad (10) \\ & E^*(A), F^*(A) = \underset{E,F}{\operatorname{argmin}} \; L(E, A, F, D^{(tr)}) \end{aligned}$$



AutoCaption: Some food is on a white plate.
IUC: A white plate of fish and broccoli is sitting on a table
GT: The meal of fish has a side of broccoli.

AutoCaption: A street sign next to a tree.
IUC: A red do not enter sign with two green street signs above it.
GT: A red do not enter sign under a green street sign.

AutoCaption: A young man is standing next to a bed.
IUC: A man with a leopard robe is standing next to a white bed.
GT: A man dressed in leopard robe next to a bed.

AutoCaption: A white refrigerator in the patio.
IUC: A dirty refrigerator and some garbage on the floor next to a building.
GT: An abandoned refrigerator next to a building with a window.

Figure 3: Exemplar captions generated by IUC and AutoCaption$^{\Sigma}$ [41] as well as their corresponding ground truth sentences generated by humans.

Table 5: Image captioning evaluation with different tradeoff parameters (λ and γ) on COCO "Karpathy" test split (single-model with Cross-Entropy Loss).

Lambda λ	Gamma γ	BLEU-4	METEOR	CIDEr	SPICE
1	0	**39.7**	**30.9**	**131.1**	**23.8**
0	1	39.1	29.7	129.9	22.9
1	1	39.5	30.4	130.6	23.1

Table 6: Comparison of utilizing various differentiable architecture search based methods with IUC on the COCO "Karpathy" test split (single-model with Cross-Entropy Loss).

Methods	BLEU-4	METEOR	CIDEr	SPICE
IUC-E + DARTS	39.7	30.9	131.1	23.8
IUC-E + PC-DARTS	39.8	31.2	131.6	**24.1**
IUC-E + DATA	**40.1**	**31.3**	**131.9**	23.9

Table 5 demonstrates the performance of IUC-E in three different cases: **1)** $\lambda = 1$ and $\gamma = 0$, the encoder-decoder model updates its architecture by minimizing the validation loss of the predictive model only, without considering the validation loss of itself - similar to Eq.(4) - and this model achieves the best performance on all four metrics. **2)** $\lambda = 0$ and $\gamma = 1$. Unlike the first case, we have a bi-level optimization problem since the encoder-decoder model updates its architecture by minimizing its validation loss without going through the second stage. This model exhibits the lowest performance. **3)** When $\lambda = 1$ and $\gamma = 1$, we are combining the validation loss of the predictive model and encoder-decoder model. In this scenario, we can achieve higher performance than in the second case since the predictive model provides more useful feedback, which assists in better learning. The achievement of these three cases implies the significant impact of $L(W^*(E^*(A), F^*(A)), D^{(val)})$ in the validation loss, which helps the model to improve its understanding of the images.

Ablation 3. In spite of the high achievements of DARTS [22], some of the newer variations of differentiable NAS methods were able to enhance the performance of DARTS and reduce its memory cost by utilizing different techniques on DARTS.
To study some of these methods for additional enhancements, we evaluate our proposed models in image captioning by applying various DARTS-based methods, including DARTS, PC-DARTS, and DATA to search for the image encoder architecture of the IUC-E model. Evaluation results in Table 6 show that utilizing more advanced DARTS-based methods, such as DATA, can be applied for further improvements on top of IUC.

5 CONCLUSION

In mission-critical applications (e.g., disease diagnosis), if the textual descriptions generated by image captioning models are incorrect, they may mislead human decision-makers and have potential negative social impacts. Thus, it is crucial to minimize the prediction error in such tasks. To tackle this issue, we presented a novel approach for image captioning tasks by utilizing differentiable NAS techniques to obtain the high-performance encoder-decoder model architectures. We introduced a three-level optimization problem by formulating IUC and provided efficient solutions to this specific framework. Furthermore, our investigations show the effectiveness of the encoder and decoder modules individually in image understanding. We applied our proposed methods on COCO image captions dataset to verify IUC can outperform the existing state-of-the-art methods in the image captioning tasks.

REFERENCES

[1] Peter Anderson, Basura Fernando, Mark Johnson, and Stephen Gould. 2016. Spice: Semantic propositional image caption evaluation. In *European conference on computer vision*. Springer, 382–398.

[2] Peter Anderson, Xiaodong He, Chris Buehler, Damien Teney, Mark Johnson, Stephen Gould, and Lei Zhang. 2018. 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*. 6077–6086.

[3] Viktar Atliha and Dmitrij Šešok. 2022. Image-Captioning Model Compression. *Applied Sciences* 12, 3 (2022), 1638.

[4] Satanjeev Banerjee and Alon Lavie. 2005. METEOR: An Automatic Metric for MT Evaluation with Improved Correlation with Human Judgments. In *Proceedings of the ACL Workshop on Intrinsic and Extrinsic Evaluation Measures for Machine Translation and/or Summarization*. Association for Computational Linguistics, Ann Arbor, Michigan, 65–72. https://aclanthology.org/W05-0909

[5] Jianlong Chang, Yiwen Guo, GAOFENG MENG, SHIMING XIANG, Chunhong Pan, et al. 2019. Data: Differentiable architecture approximation. *Advances in Neural Information Processing Systems* 32 (2019), 876–886.

[6] Kyunghyun Cho, Aaron Courville, and Yoshua Bengio. 2015. Describing multimedia content using attention-based encoder-decoder networks. *IEEE Transactions on Multimedia* 17, 11 (2015), 1875–1886.

[7] Marcella Cornia, Matteo Stefanini, Lorenzo Baraldi, and Rita Cucchiara. 2020. Meshed-memory transformer for image captioning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 10578–10587.

[8] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. 2009. Imagenet: A large-scale hierarchical image database. In *2009 IEEE conference on computer vision and pattern recognition*. Ieee, 248–255.

[9] Bhanu Garg, Li Zhang, Pradyumna Sridhara, Ramtin Hosseini, Eric Xing, and Pengtao Xie. 2021. Learning from Mistakes–A Framework for Neural Architecture Search. *arXiv preprint arXiv:2111.06353* (2021).

[10] Yu-Chao Gu, Li-Juan Wang, Yun Liu, Yi Yang, Yu-Huan Wu, Shao-Ping Lu, and Ming-Ming Cheng. 2021. Dots: Decoupling operation and topology in differentiable architecture search. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 12311–12320.

[11] Ramtin Hosseini and Pengtao Xie. 2020. Learning by Self-Explanation, with Application to Neural Architecture Search. *arXiv preprint arXiv:2012.12899* (2020).

[12] Ramtin Hosseini, Xingyi Yang, and Pengtao Xie. 2021. DSRNA: Differentiable Search of Robust Neural Architectures. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 6196–6205.

[13] Lun Huang, Wenmin Wang, Jie Chen, and Xiao-Yong Wei. 2019. Attention on attention for image captioning. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*. 4634–4643.

[14] Wenhao Jiang, Lin Ma, Yu-Gang Jiang, Wei Liu, and Tong Zhang. 2018. Recurrent fusion network for image captioning. In *Proceedings of the European Conference on Computer Vision (ECCV)*. 499–515.

[15] Andrej Karpathy and Li Fei-Fei. 2015. Deep visual-semantic alignments for generating image descriptions. In *Proceedings of the IEEE conference on computer vision and pattern recognition*. 3128–3137.

[16] Diederik P Kingma and Jimmy Ba. 2014. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980* (2014).

[17] Ranjay Krishna, Yuke Zhu, Oliver Groth, Justin Johnson, Kenji Hata, Joshua Kravitz, Stephanie Chen, Yannis Kalantidis, Li-Jia Li, David A Shamma, et al. 2017. Visual genome: Connecting language and vision using crowdsourced dense image annotations. *International journal of computer vision* 123, 1 (2017), 32–73.

[18] Xiujun Li, Xi Yin, Chunyuan Li, Pengchuan Zhang, Xiaowei Hu, Lei Zhang, Lijuan Wang, Houdong Hu, Li Dong, Furu Wei, et al. 2020. Oscar: Object-semantics aligned pre-training for vision-language tasks. In *European Conference on Computer Vision*. Springer, 121–137.

[19] Chin-Yew Lin. 2004. Rouge: A package for automatic evaluation of summaries. In *Text summarization branches out*. 74–81.

[20] Tsung-Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C Lawrence Zitnick. 2014. Microsoft coco: Common objects in context. In *European conference on computer vision*. Springer, 740–755.

[21] Hanxiao Liu, Karen Simonyan, Oriol Vinyals, Chrisantha Fernando, and Koray Kavukcuoglu. 2018. Hierarchical Representations for Efficient Architecture Search. In *ICLR*.

[22] Hanxiao Liu, Karen Simonyan, and Yiming Yang. 2018. Darts: Differentiable architecture search. *arXiv preprint arXiv:1806.09055* (2018).

[23] Junhua Mao, Wei Xu, Yi Yang, Jiang Wang, and Alan L Yuille. 2014. Explain images with multimodal recurrent neural networks. *arXiv preprint arXiv:1410.1090* (2014).

[24] Yingwei Pan, Ting Yao, Yehao Li, and Tao Mei. 2020. X-linear attention networks for image captioning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 10971–10980.

[25] 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*. 311–318.

[26] Hieu Pham, Melody Y. Guan, Barret Zoph, Quoc V. Le, and Jeff Dean. 2018. Efficient Neural Architecture Search via Parameter Sharing. In *ICML*.

[27] Yu Qin, Jiajun Du, Yonghua Zhang, and Hongtao Lu. 2019. Look back and predict forward in image captioning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 8367–8375.

[28] Esteban Real, Alok Aggarwal, Yanping Huang, and Quoc V. Le. 2019. Regularized Evolution for Image Classifier Architecture Search. *Proceedings of the AAAI Conference on Artificial Intelligence* 33, 01 (July 2019), 4780–4789. https://doi.org/10.1609/aaai.v33i01.33014780 Number: 01.

[29] Shaoqing Ren, Kaiming He, Ross Girshick, and Jian Sun. 2015. Faster r-cnn: Towards real-time object detection with region proposal networks. *Advances in neural information processing systems* 28 (2015), 91–99.

[30] Steven J Rennie, Etienne Marcheret, Youssef Mroueh, Jerret Ross, and Vaibhava Goel. 2017. Self-critical sequence training for image captioning. In *Proceedings of the IEEE conference on computer vision and pattern recognition*. 7008–7024.

[31] Ramakrishna Vedantam, C Lawrence Zitnick, and Devi Parikh. 2015. Cider: Consensus-based image description evaluation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*. 4566–4575.

[32] Oriol Vinyals, Alexander Toshev, Samy Bengio, and Dumitru Erhan. 2015. Show and tell: A neural image caption generator. In *Proceedings of the IEEE conference on computer vision and pattern recognition*. 3156–3164.

[33] Yuhui Xu, Lingxi Xie, Xiaopeng Zhang, Xin Chen, Guo-Jun Qi, Qi Tian, and Hongkai Xiong. 2019. PC-DARTS: Partial channel connections for memory-efficient architecture search. *arXiv preprint arXiv:1907.05737* (2019).

[34] Yuhui Xu, Lingxi Xie, Xiaopeng Zhang, Xin Chen, Guo-Jun Qi, Qi Tian, and Hongkai Xiong. 2020. {PC}-{DARTS}: Partial Channel Connections for Memory-Efficient Architecture Search. In *International Conference on Learning Representations*. https://openreview.net/forum?id=BJlS634tPr

[35] Xu Yang, Kaihua Tang, Hanwang Zhang, and Jianfei Cai. 2019. Auto-encoding scene graphs for image captioning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 10685–10694.

[36] Ting Yao, Yingwei Pan, Yehao Li, and Tao Mei. 2018. Exploring visual relationship for image captioning. In *Proceedings of the European conference on computer vision (ECCV)*. 684–699.

[37] Ting Yao, Yingwei Pan, Yehao Li, Zhaofan Qiu, and Tao Mei. 2017. Boosting image captioning with attributes. In *Proceedings of the IEEE international conference on computer vision*. 4894–4902.

[38] Quanzeng You, Hailin Jin, Zhaowen Wang, Chen Fang, and Jiebo Luo. 2016. Image captioning with semantic attention. In *Proceedings of the IEEE conference on computer vision and pattern recognition*. 4651–4659.

[39] Le Zhang, Yanshuo Zhang, Xin Zhao, and Zexiao Zou. 2021. Image captioning via proximal policy optimization. *Image and Vision Computing* 108 (2021), 104126.

[40] Pengchuan Zhang, Xiujun Li, Xiaowei Hu, Jianwei Yang, Lei Zhang, Lijuan Wang, Yejin Choi, and Jianfeng Gao. 2021. Vinvl: Revisiting visual representations in vision-language models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 5579–5588.

[41] Xinxin Zhu, Weining Wang, Longteng Guo, and Jing Liu. 2020. AutoCaption: Image Captioning with Neural Architecture Search. *arXiv preprint arXiv:2012.09742* (2020).

[42] Barret Zoph and Quoc V Le. 2016. Neural architecture search with reinforcement learning. *arXiv preprint arXiv:1611.01578* (2016).

[43] Barret Zoph, Vijay Vasudevan, Jonathon Shlens, and Quoc V Le. 2018. Learning transferable architectures for scalable image recognition. In *CVPR*.