# METHODS TO IMPROVE QUALITY AND DIVERSITY OF LANGUAGE-VISION MODELS

BY

JYOTI ANEJA

DISSERTATION

Submitted in partial fulfillment of the requirements
for the degree of Doctor of Philosophy in Physics
in the Graduate College of the
University of Illinois Urbana-Champaign, 2021

Urbana, Illinois

Doctoral Committee:

Associate Professor Benjamin Hooberman, Chair
Assistant Professor Alexander G. Schwing, Director of Research
Professor Lance Cooper
Professor David Forsyth
Professor Svetlana Lazebnik

## ABSTRACT

Humans can describe images and, more generally, the world around them in an evocative manner using vivid language constructs. Designing neural network models that can attain results similar to those of humans on tasks like image-captioning and image-generation is a worthy goal in the overall pursuit of artificial general intelligence. Notwithstanding the tremendous recent progress in this area, current systems still cannot describe objects and scenes as creatively and accurately as humans. As a step in the direction of bridging this gap, this thesis proposes architectures and algorithms for generating high-quality, diverse outputs for the tasks of image-captioning and image-generation.

We start with *ConvCap*, which replaces the LSTM network used in prior image-captioning generators with masked convolutions. We analyze the network characteristics to obtain a deeper understanding of the differences between recurrent and convolutional architectures. We further show that our approach is faster to train as the convolutions are amenable to parallelization. Training captioning models that produce varied outputs instead of regurgitating the training data is an important challenge. Towards that, we introduce the *PosCap* and *Seq-CVAE* models. The former infuses diversity by using part-of-speech tags to condition the generation of captions. In the latter, we switch to latent variable models (VAEs) and design an approach to garner diversity by using hierarchical latent variables for the different words in a caption. Then, in *DivCap*, we formulate image captioning as a multi-objective optimization problem and strive to obtain a balance between the accuracy and the diversity of the generated captions. We conclude with *NCP-VAE* for high-quality image generation with hierarchical VAEs. Specifically, our focus is to tackle the mismatch between the prior and the aggregate approximate posterior distributions in a VAE (referred to as the prior hole problem).

*To my parents,*
*Amrit and Ashok Aneja,*
*for their limitless trust in my passion and eternal support in my endeavour,*
*and to my siblings,*
*Tripti and Keshav,*
*for their love and care through this often arduous process.*

## ACKNOWLEDGMENTS

This dissertation would not have been possible without the support of several amazing people. First and foremost I am grateful to my thesis advisor Professor Alexander Schwing. Alex is a remarkably inspiring person with immense passion and dedication for his work and the success of his students. He has always been extremely patient and generous with his time. I learned the art of research from Alex. He not only agreed to work with me (a graduate student from the physics department) but often accentuated the value of taking a non-traditional path. In the last few years in Alex, I found a fantastic mentor, a great collaborator, and a reliable friend.

Next, I thank my doctoral committee members Professor Benjamin Hooberman, Professor Lance Cooper, Professor David Forsyth and Professor Svetlana Lazebnik. First two works in this thesis were a culmination of a wonderful collaboration with David. The project discussion meetings, whenever they happened, were the highlight of the day. David's clarity of thought and succinctness of language are only second to his own research genius. Early on in my effort to get familiar with machine learning and deep learning, I took the fantastic 'Cutting-Edge Trends in Deep Learning and Recognition' course offered by Lana. This course provided a comprehensive review of where the field is going and where it was coming from. This helped me put things in perspective and understand what I wanted to work on. I also had the fortune of being a TA with Lana in my last semester, and my only regret is that I could not get a chance to work with her on a research project.

All PhDs are extremely personal and unique journeys of learning and growth, mine was too. I completely switched directions from physics to computer science much later in the program. This move would not have been possible without the perpetual support from Professor Lance Cooper. I offer my deepest gratitude to Lance for not only supporting me every step of the way in navigating a complicated path that I had chosen but also opening several doors which I never thought would have been possible.

Articles in this thesis are a joint effort with my collaborators. I was fortunate to work with these fantastic people - Aditya Deshpande, Harsh Agrawal (Georgia Tech), and Xiaoming Zhao. A special mention to Harsh for his friendship and often being the source of encouragement and guidance. Parts of this thesis were completed through internship collaborations. Thanks to Ziyu Zhang (internship at Snap), Neel Joshi (internship at Microsoft Research), and Arash Vahdat (internship at Nvidia) for the enjoyable and technically fulfilling collaboration experiences.

My years at UIUC have been incredibly enriching and enjoyable thanks to many graduate students and friends from the Physics and Computer Science departments. Maghav Kumar, Aditya Deshpande, Anand Bhattad, Unnat Jain, Tanmay Gupta, Rajbir Kataria, Zhizhong Li, Daniel McKee, Arun Mallya, Kanika Narang, Ketan Mittal, YuanTing Hu, Safa Messaoud, Zhongzheng Ren, Raymond Yeh, Ashok Makkuva, Matt Zhang, Pulkit Budhiraja, Aditi Khullar, Gaurav Singh, Xiaoming Zhao, Sushmita Das.

Special thanks to my partner Tanmay Gangwani for his comradeship and care through the years. I look forward to our lives ahead.

Finally, there are no words enough to express my gratitude towards my family. My parents for their unwavering love, encouragement, and support. They taught me to be positive in uncertainty and brave in adversity. Without their values and sacrifices, I would not be where I am today. My siblings, who have stood by me through everything in life. They had immense trust in my decisions and capabilities and this thesis is dedicated to them.

**TABLE OF CONTENTS**

# CHAPTER 1: THESIS OVERVIEW

Efficiently describing images, objects, and scenes is a long-standing task in artificial intelligence. The complexity of the objective stems from the fact that to achieve success, the AI model must learn excellent visual perception and recognition skills, along with meaningful representations of real-world concepts and natural language constructs. Furthermore, they need to develop strong vision-language understanding by correlating representations across these two modalities. While humans can describe their worlds creatively, quickly, and with remarkable ease, designing machine learning models to imitate this behavior is still an open challenge. **To that end, in this thesis, we introduce architectures and algorithms for generating high-quality, diverse outputs for the tasks of image-captioning and image-generation.**

## 1.1 MAIN CONTRIBUTIONS

We now expand on some of the main challenges in the domain of image-captioning and image-generation. We provide summaries of our proposed approaches to tackle those challenges, and discuss them in detail in the following chapters. Section 1.1.1 focuses on building convolutional models for image captioning as a replacement for the recurrent network used in prior literature. In Section 1.1.2 and Section 1.1.3, we discuss approaches to generate diverse captions for a given image using part-of-speech tag information and sequential latent-variable models, respectively. In Section 1.1.4, we discuss the multi-objective optimization approach to balance accuracy and diversity metrics for image captioning. Lastly, Section 1.1.5 discusses a solution to overcome the prior hole problem faced by variational autoencoders, popular for generating diverse high-quality image and language outputs.

### 1.1.1 Convolutional Models for Image Captioning (CVPR-2018)

Recurrent Networks like Long Short Term Memory networks (LSTMs) have been considered the de-facto standard for vision-language tasks of image captioning, visual question answering, question generation, and visual dialog, due to their compelling ability to memorize long-term dependencies through a memory cell. However, the complex addressing and overwriting mechanism combined with inherently sequential processing, and significant storage required due to back-propagation through time (BPTT), poses challenges during training. Also, in contrast to convolutional neural networks (CNNs), that are non-sequential, LSTMs

often require more careful engineering, when considering a novel task. Previously, CNNs have not matched up to the LSTM performance on vision-language tasks. We study convolutional architectures for the vision-language task of image captioning. Our approach has comparable performance to LSTM based methods. An attention mechanism leveraging spatial image features improves performance. CNNs produce more entropy (useful for diverse predictions), better classification accuracy, and do not suffer from vanishing gradients (Aneja et al., 2018).

### 1.1.2 Image Captioning using Part-of-Speech as Prior (CVPR-2019)

We enable an image captioning system to generate diverse captions by conditioning on different high-level summaries of the image. Our summaries are quantized part-of-speech (POS) tag sequences. Our system generates captions by (a) predicting different summaries from the image, then (b) predicting captions conditioned on each summary. This approach leads to captions that are *accurate, quick to obtain*, and *diverse*. Beam search is the de-facto standard method for sampling multiple captions. Our system is accurate because it can steer several narrow beam searches to explore the space of caption sequences more efficiently. It is fast because each beam is narrow. And the captions are diverse because, depending on the summary (*i.e.*, POS), the system is forced to produce captions that contain (for example) more or fewer adjectives. This means we avoid the tendency to produce minimal or generic captions that is common in systems that try to optimize likelihood without awareness of language priors (like POS) (Deshpande et al., 2019).

### 1.1.3 Sequential Latent Space Models for Diverse Image Captioning (ICCV-2019)

A popular approach to promote output-space diversity in vision-language tasks is to use latent-variable models, such as a VAE. However, in prior works with such models, the latent variable either only initializes the sentence generation process, or is identical across the steps of generation. Hence they do not offer fine-grained control over the generation process. To address this concern, we propose Seq-CVAE which learns a latent space for every word of the sentence. We encourage this temporal latent space to capture the *intention* about how to complete the sentence by mimicking a representation that summarizes the future. This approach also allows the use of the entire sentence at once, at training time, contrary to all the previous approaches which use the past words to train the future (Aneja et al., 2019).

#### 1.1.4 Formulating Image Captioning as a Multi-Objective Optimization Problem (under-submission)

Image captioning is challenging because of its two competing goals: accuracy and diversity. The former means that the captions should be consistent with the image and language rules; the latter is desired since an image can be described in a myriad of ways. Existing methods that perform well on perceptual accuracy metrics often don't cater to the fact that captioning is ambiguous. Conversely, existing methods that achieve diverse results often suffer from much lower accuracy. Additionally, previous approaches either use explicit conditioning (Deshpande et al., 2019) or learn an implicit latent space (Aneja et al., 2019; Wang et al., 2017c) to condition the generation of diverse captions. In this work, we propose to formulate image captioning as a multi-objective optimization and strive to obtain Pareto optimality. On the MSCOCO captioning dataset, we observe that this yields much more balanced image descriptions: they are accurate, yet the model is able to cater to the inherent ambiguity.

#### 1.1.5 A Contrastive Learning Approach for Training Variational Autoencoder Priors (NeurIPS-2021)

One common observation from training VAE-based models is the *prior-hole problem*: the VAE prior (used for test-time image generation) fails to match the aggregate approximate posterior from the training phase. This constitutes part of the explanation for VAEs' poor generative quality. Due to this mismatch, there exist areas in the latent space with high density under the prior that do not correspond to any encoded image. Samples from those areas are decoded to corrupted images. To tackle this issue, we propose an energy-based prior defined by the product of a base prior distribution and a re-weighting factor, designed to bring the base closer to the aggregate posterior. Our current approach provides state-of-the-art results on the task of VAE-based image generation. In the future, we plan to extend the idea to other vision-language tasks (Aneja et al., 2021).

### 1.2 THESIS ORGANIZATION

This thesis is organized in the following three parts:

**Part I: Architectural Enhancements for Language-Vision Models** consists of 1 chapter. Chapter 2 proposes a completely convolutional approach to train image captioning models.

**Part II: Learning Models that Generate Diverse Outputs in a Computationally Efficient Manner** consists of three chapters. Chapters 3 and 4 propose methods for explicitly and implicitly conditioning the captioning system to generate diverse outputs, respectively. Chapter 5 employs multi-objective optimization to achieve the accuracy-diversity balance.

**Part III: Tackling Prior and Aggregate-Posterior Mismatch in Variational Autoencoders** consists of 1 chapter. Chapter 6 considers using noise contrastive estimation to alleviate the prior-hole problem of variational autoencoders.

# Part I

# Architectural Enhancements for Language-Vision Models

# CHAPTER 2: CONVOLUTIONAL IMAGE CAPTIONING

## 2.1 INTRODUCTION

Image captioning, *i.e.*, describing the content observed in an image, has received a significant amount of attention in recent years. It is applicable in various scenarios, *e.g.*, recommendation in editing applications, usage in virtual assistants, for image indexing, and support of the disabled. With the availability of large datasets, deep neural network (DNN) based methods have been shown to achieve impressive results on image captioning tasks (Karpathy & Fei-Fei, 2015; Vinyals et al., 2015a). These techniques are largely based on recurrent neural nets (RNNs), often powered by a Long-Short-Term-Memory (LSTM) (Hochreiter & Schmidhuber, 1997b) component. LSTM nets have been considered as the de-facto standard for vision-language tasks of image captioning (Chen & Zitnick, 2015; Karpathy & Fei-Fei, 2015; Vinyals et al., 2015a; Xu et al., 2015; Wang et al., 2017a), visual question answering (Antol et al., 2015; Shih et al., 2016; Schwartz et al., 2017), question generation (Jain et al., 2017; Mostafazadeh et al., 2016), and visual dialog (Das et al., 2017; Jain et al., 2018), due to their compelling ability to memorize long-term dependencies through a memory cell. However, the complex addressing and overwriting mechanism combined with inherently sequential processing, and significant storage required due to back-propagation through time (BPTT), poses challenges during training. Also, in contrast to CNNs, that are non-sequential, LSTMs often require more careful engineering, when considering a novel task. Previously, CNNs have not matched up to the LSTM performance on vision-language tasks. Inspired by the recent successes of convolutional architectures on other sequence-to-sequence tasks – conditional image generation (van den Oord et al., 2016), machine translation (Gehring et al., 2017; Vaswani et al., 2017) – we study convolutional architectures for the vision-language task of image captioning. To the best of our knowledge, ours is the first convolutional network for image captioning that compares favorably to LSTM-based methods.

Our key contributions are: **a)** A convolutional (CNN-based) image captioning method that shows comparable performance to an LSTM based method (Karpathy & Fei-Fei, 2015) (Section 2.6.2, Table 2.1 and Table 2.2); **b)** Improved performance with a CNN model that uses attention mechanism to leverage spatial image features. With attention, we outperform the attention baseline (Xu et al., 2015) and qualitatively demonstrate that our method finds salient objects in the image. (Figure 2.5, Table 2.2); **c)** We analyze the characteristics of CNN and LSTM nets and provide useful insights such as – CNNs produce more entropy (useful for diverse predictions), better classification accuracy, and do not suffer from vanishing gradients

(Section 2.6 and Figure 2.6, 2.7 and 2.8). We evaluate our architecture on the challenging MSCOCO (Lin et al., 2014) dataset, and compare it to an LSTM (Karpathy & Fei-Fei, 2015) and an LSTM+Attention baseline (Xu et al., 2015).

## 2.2 PROBLEM SETUP AND NOTATION

For image captioning, we are given an input image $I$ and we want to generate a sequence of words $y = (y_1, \ldots, y_N)$. The possible words $y_i$ at time-step $i$ are subsumed in a discrete set $\mathcal{Y}$ of options. Its size, $|\mathcal{Y}|$, easily reaches several thousands. $\mathcal{Y}$ contains special tokens that denote a start token (<S>), an end of sentence token (<E>), and an unknown token (<UNK>) which refers to all words not in $\mathcal{Y}$.

Given a training set $\mathcal{D} = \{(I, y^*)\}$ which contains pairs $(I, y^*)$ of input image $I$ and corresponding ground-truth caption $y^* = (y_1^*, \ldots, y_N^*)$, consisting of words $y_i^* \in \mathcal{Y}$, $i \in \{1, \ldots, N\}$, we maximize w.r.t. parameters $w$, a probabilistic model $p_w(y_1, \ldots, y_N|I)$. A variety of probabilistic models have been considered (Section 2.7), from hidden Markov models (Yang et al., 2011) to recurrent neural networks. Note, in the subsequent sections we use RNN and LSTM interchangeably. As prevalent in the community, these are often used as a replacement for one another. The difference lies in the fact that RNN is a recurrent neural network architecture and LSTM is the (most popularly used) cell powering this architecture.

## 2.3 RNN APPROACH

An illustration of a classical RNN architecture for image captioning is provided in Figure 2.1. It consists of three major components, all of which contain trainable parameters: the input word embeddings, the sequential LSTM units containing the memory cell, and the output word embeddings.

**Inference.** RNNs sequentially predict one word at a time, from $y_1$ up to $y_N$. At every time-step $i$, a conditional probability distribution $p_{i,w}(y_i|h_i, I)$, which depends on parameters $w$, is predicted (see top of Figure 2.1). For modeling $p_{i,w}(y_i|h_i, I)$, in the spirit of auto-regressive models, the dependence of word $y_i$ on its ancestors $y_{<i}$ is implicitly captured by a hidden representation $h_i$ (see arrows in Figure 2.1). Formally, the probability is computed via

$$p_{i,w}(y_i|h_i, I) = g_w(y_i, h_i, I), \tag{2.1}$$

where $g_w$ can be any differentiable function/deep net. Note, image captioning techniques usually encode the image into the hidden representation $h_0$ (Figure 2.1).

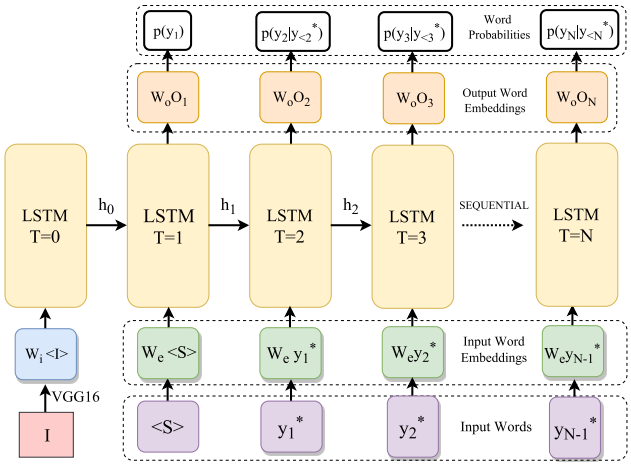



Figure 2.1: A sequential RNN powered by an LSTM cell. At each time step the output is conditioned on the previously generated word, the image is fed at the start only. Feature vector for image $I$, coming from an image encoder (VGG16) is fed as input through the image-embedding $W_i$ at the $0^{th}$ time step. Start token $< S >$ is then fed through the word-embedding $W_e$ at the first time step. This makes the training and inference setting consistent. Subsequently ground-truth words $y_i^*$ are fed through the word-embedding $W_e$ at the $i^{th}$ time step, where, $i \in \{2, \ldots, N\}$.

Importantly, RNNs are described by a recurrence relation which governs computation of the hidden state $h_i$ via

$$h_i = f_w(h_{i-1}, y_{i-1}, I). \tag{2.2}$$

Again, $f_w$ can be any differentiable function. For image captioning, long-short-term-memory (LSTM) (Hochreiter & Schmidhuber, 1997b) nets and variants thereof based on gated recurrent units (GRU) (Cho et al., 2014), or forward-backward LSTM nets are used here.

**Learning.** Following classical supervised learning, it is common to find the parameters $w$ of the word embeddings and the LSTM unit by minimizing the negative log-likelihood of the training data $\mathcal{D}$, *i.e.*, we optimize:

$$\min_w \sum_{\mathcal{D}} \sum_{i=1}^{N} -\ln p_{i,w}(y_i^*|h_i, I). \tag{2.3}$$

To compute the gradient of the objective given in Eq. (2.3), we use back-propagation through time (BPTT). BPTT is necessary due to the recurrence relationship encoded in $f_w$ (Eq. (2.2)). Note, the gradients of the function $f_w$ at time $i$ depend on the gradients obtained in successive time-steps.

To avoid more complicated gradient flows through the recurrence relationship, during

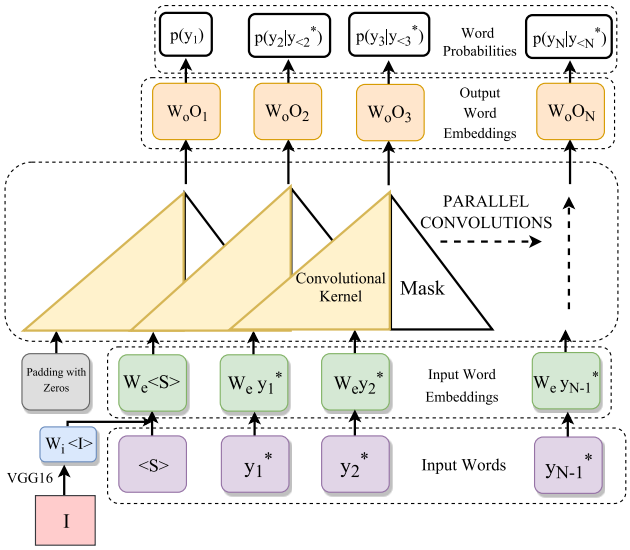



Figure 2.2: Our convolutional model for image captioning. We use a feed forward network with masked convolutions. Unlike RNNs, our model operates over all words in parallel.

training, it is common to use

$$h_i = f_w(h_{i-1}, y^*_{i-1}, I), \tag{2.4}$$

rather than the form provided in Eq. (2.2). *I.e.*, during training, when computing the latent representation $h_i$, we use the ground-truth symbol $y^*_{i-1}$ rather than the prediction $y_{i-1}$. This is termed as teacher forcing.

Although highly successful, RNN-based techniques suffer from some drawbacks. First, the training process is inherently sequential for a particular image-caption pair. This results from unrolling the recurrent relation in time. Hence, the output at time-step $i$ has a true dependency on the output at $i-1$. Secondly, as we will show in our results for image captioning, RNNs tend to produce lower classification accuracy (Figure 2.6), and, despite LSTM units, they still suffer to some degree from vanishing gradients (Figure 2.8).

Next, we describe an alternative convolutional approach to image captioning which attempts to overcome some of these challenges.

## 2.4 CONVOLUTIONAL APPROACH

Our model is based on the convolutional machine translation model used by Gehring et al. (2017). Figure 2.2 provides an overview of our feed-forward convolutional (or CNN-based) approach for image captioning. As the figure illustrates, our technique contains three main components similar to the RNN technique. The first and the last components are

input/output word embeddings respectively, in both cases. However, while the middle component contains LSTM or GRU units in the RNN case, masked convolutions are employed in our CNN-based approach. This component, unlike the RNN, is feed-forward without any recurrent function. We briefly review inference and learning of our model.

**Inference:** In contrast to the RNN formulation, where the probabilistic model is unrolled in time via the recurrence relation given in Eq. (2.2), we use a simple feed-forward deep net, $f_w$, for modeling $p_{i,w}(y_i|I)$. Prediction of a word $y_i$ relies on past words $y_{<i}$ or their representations:

$$p_{i,w}(y_i|y_{<i}, I) = f_w(y_i, y_{<i}, I). \tag{2.5}$$

To disallow convolution operations from using information of future word tokens, we use masked convolutional layers that operate only on 'past' data (Gehring et al., 2017; van den Oord et al., 2016).

Inference can now be performed sequentially, one word at a time. Hence, inference begins with the start token <S> and employs a feed-forward pass to generate $p_{1,w}(y_1|\emptyset, I)$. Afterwards, $y_1 \sim p_{1,w}(y_1|\emptyset, I)$ is sampled. Note that it is possible to retrieve the maximizing argument or to perform beam search. After sampling, $y_1$ is fed back into the feed-forward network to generate subsequent words $y_2$, *etc*. Inference continues until the end token is predicted, or until we reach a fixed upper bound of $N$ steps.

**Learning:** Similar to RNN training, we use ground-truth $y^*_{<i}$ for past words, instead of using the predicted word. For prediction of word probability $p_{i,w}(y_i|y^*_{<i}, I)$, the considered feed-forward network is $f_w(y_i, y^*_{<i}, I)$ and we optimize for parameters $w$ using a likelihood similar to Eq. (2.3).

Since there are no recurrent connections and all ground-truth words are available at any given time-step $i$, our CNN based model can be trained in parallel for all words. In Section 2.5, we describe our convolutional architecture in detail.

## 2.5 ARCHITECTURE

In Figure 2.3, we show a training iteration of our convolutional architecture with input (ground-truth) words $\{y^*_1, \ldots, y^*_5\} = \{$ a, woman, is, playing, tennis $\}$. Additionally, we add the start token <S> at the beginning, and also the end of sentence token <E>.

These words are processed as follows: (1) they pass through an input embedding layer; (2) they are combined with the image embedding; (3) they are processed by the CNN module; and (4) the output embedding (or classification) layer produces output probability distributions (see $\{p_1, \ldots, p_6\}$ at top of Figure 2.3). Each of the four aforementioned steps

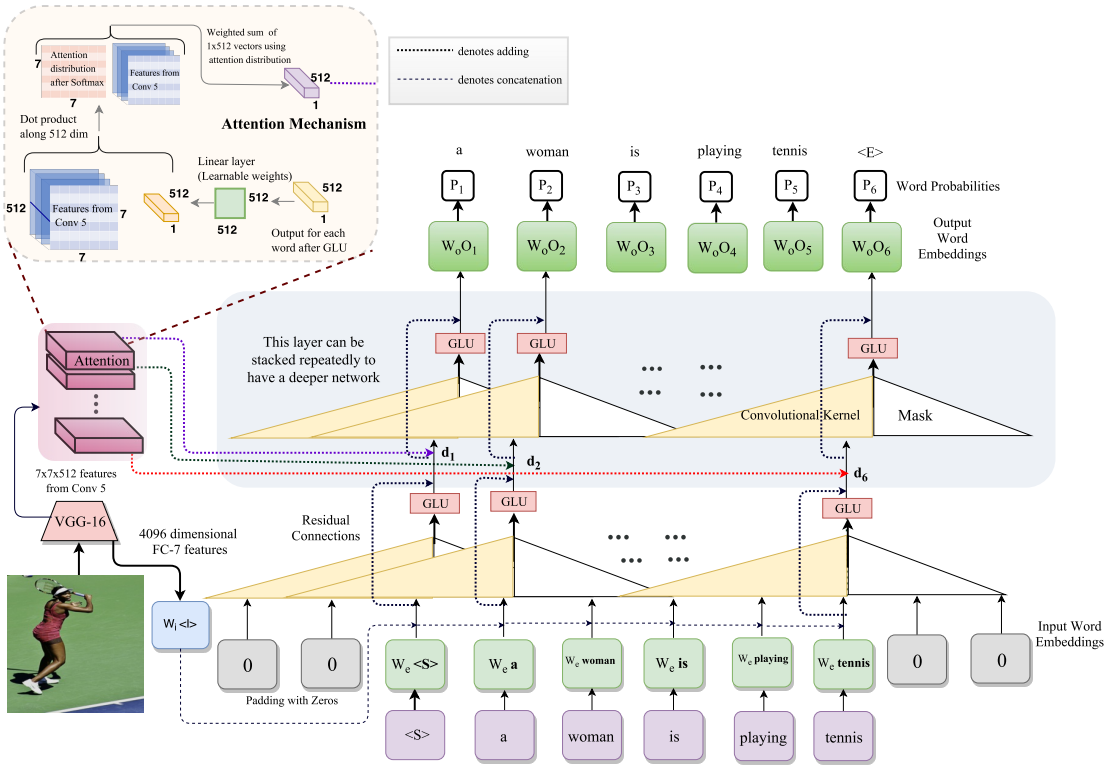



Figure 2.3: Our convolutional architecture for image captioning. It has four components: (i) Input embedding layer, (ii) Image embedding, (iii) Convolutional module and (iv) Output embedding layer. Details of each component are in Section 2.5.

is discussed below.

**Input Embedding:** For consistency with the RNN/LSTM baseline, we train (from scratch) an embedding layer over one-hot encoded input words. We use $|\mathcal{Y}|= 9221$ and we embed the input words to 512-dimensional vectors, following the baseline. This embedding is concatenated to the image embedding (discussed next) and provided as input to the feed-forward CNN module.

**Image Embedding:** Image features for image $I$ are obtained from the fc7 layer of the VGG16 network (Simonyan & Zisserman, 2015). The VGG16 is pre-trained on the ImageNet dataset (Russakovsky et al., 2015). We apply dropout, ReLU on the fc7 and use a linear layer to obtain a 512-dimensional embedding. This is consistent with the image features used in the baseline LSTM method (Karpathy & Fei-Fei, 2015).

**CNN Module:** The CNN module operates on the combined input and image embedding vector. It performs three layers of masked convolutions. Consistent with (Gehring et al., 2017; van den Oord et al., 2016), we use gated linear unit (or GLU) activations for our conv layers. However, we did not observe a significant change in performance when using the standard ReLU activation. The feature dimension after convolution layer and GLU is 512.

We add weight normalization, residual connections and dropout in these layers as they help improve performance (Table 2.1). Our masked convolutions have a receptive field of 5 words in the past. We set $N$ (steps or max-sentence length) to 15 for both CNN/RNN. The output of the CNN module after three layers is a 512-dimensional vector for each word.

**Classification Layer:** We use a linear layer to encode the 512-dimensional vectors obtained from the CNN module into a 256-dimensional representation per word. Then, we upsample this vector to a $|\mathcal{Y}|$-dimensional activation via a fully connected layer, and pass it through a softmax to obtain the output word probabilities $p_{i,w}(y_i|y_{<i}, I)$.

**Training:** We use a cross-entropy loss on the probabilities $p_{i,w}(y_i|y_{<i}, I)$ to train the CNN module and the embedding layers. Consistent with (Karpathy & Fei-Fei, 2015), we start to fine-tune VGG16 along with our network after 8 training epochs. We optimize with RMSProp using an initial learning rate of $5e^{-5}$ and decay it by multiplying with a factor of .1 every 15 epochs. All methods were trained for 30 epochs and we evaluate the metrics (in Section 2.6.2) on the validation set, after every epoch, to pick the best model for all methods.

### 2.5.1 Attention

In addition to the aforementioned CNN architecture, we also experiment with an attention mechanism, since attention benefited (Gehring et al., 2017; Vaswani et al., 2017). We form an attended image vector of dimension 512 and add it to the word embedding at every layer (shown with red, green and blue arrows in Figure 2.3). We compute separate attention parameters and a separate attended vector for every word. To obtain this attended vector we predict $7\times7$ attention parameters, over the VGG16 max-pooled conv-5 features of dimensions $7\times7\times512$ (Simonyan & Zisserman, 2015). We use attention on all three masked convolution layers in our CNN module. We continue to use the fc7 image embedding discussed above.

To discuss attention more formally, let $d_j$ denote the embedding of word $j$ in the conv module (*i.e.*, its activations after GLU shown in Figure 2.3), let $W$ refer to a linear layer applied to $d_j$, let $c_i$ denote a 512-dimensional spatial conv-5 feature at location $i$ (in $7 \times 7$ feature map) and let $a_{ij}$ indicate the attention parameters. With this notation at hand, the attention parameter $a_{ij}$ is computed via $a_{ij} = \frac{\exp(W(d_j)^T c_i)}{\sum_i \exp(W(d_j)^T c_i)}$, and the attended image vector for word $j$ is obtained from $\sum_i a_{ij} c_i$. Note that (Xu et al., 2015) uses the LSTM hidden state to compute the attention parameters. Instead, we compute attention parameters using the conv-layer activations. This form of attention mechanism was first proposed in (Bahdanau et al., 2014).

## 2.6 RESULTS AND ANALYSIS

| Method | MSCOCO Val Set | | | | | | | | MSCOCO Test Set | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | B1 | B2 | B3 | B4 | M | R | C | S | B1 | B2 | B3 | B4 | M | R | C | S |
| **Baselines:** | | | | | | | | | | | | | | | | |
| LSTM (Karpathy & Fei-Fei, 2015) | **.710** | .535 | .389 | .281 | **.244** | **.521** | **.899** | .169 | **.713** | **.541** | **.404** | **.303** | **.247** | **.525** | **.912** | .172 |
| LSTM + Attn (Soft) (Xu et al., 2015) | - | - | - | - | - | - | - | - | .707 | .492 | .344 | .243 | .239 | - | - | - |
| LSTM + Attn (Hard) (Xu et al., 2015) | - | - | - | - | - | - | - | - | .718 | .504 | .357 | .250 | .230 | - | - | - |
| **Our CNN:** | | | | | | | | | | | | | | | | |
| CNN | .693 | .518 | .374 | .268 | .238 | .511 | .855 | .167 | .695 | .521 | .380 | .276 | .241 | .514 | .881 | .171 |
| CNN + Weight Norm. | .702 | .528 | .384 | .279 | .242 | .517 | .881 | .169 | .699 | .525 | .382 | .276 | .241 | .516 | .878 | .170 |
| CNN +WN +Dropout | .707 | .532 | .386 | .278 | .242 | .517 | .883 | .171 | .704 | .532 | .389 | .283 | .243 | .520 | .904 | .173 |
| CNN +WN +Dropout +Residual | .706 | .532 | .389 | **.284** | **.244** | .519 | **.899** | **.173** | .704 | .532 | .389 | .284 | .244 | .520 | .906 | **.175** |
| CNN +WN +Drop. +Res. +Attn | **.710** | **.537** | **.391** | .281 | .241 | .519 | .890 | .171 | .711 | .538 | .394 | .287 | .244 | .522 | **.912** | **.175** |

Table 2.1: Comparison of different methods on standard evaluation metrics: BLEU-1 (B1), BLEU-2 (B2), BLEU-3 (B3), BLEU-4 (B4), METEOR (M), ROUGE (R), CIDEr (C) and SPICE (S). Our CNN with attention (attn) achieves comparable performance (equal CIDEr scores on MSCOCO test set) to Karpathy & Fei-Fei (2015) and outperforms LSTM+Attention baseline of Xu et al. (2015). We start with a CNN comprising masked convolutions and fully connected layers only. Then, we add weight normalization, dropout, residual connections and attention incrementally and show that performance improves with every addition. Here, for CNN and Karpathy & Fei-Fei (2015) we use the model that obtains the best CIDEr scores on val-set (over 30 epochs) and report its scores for the test set. For Xu et al. (2015), we report all the available metrics for soft/hard attention from their paper (missing numbers are marked by -).

In this section, we demonstrate the following results:

- Our convolutional (or CNN) approach performs on par with LSTM (or RNN) based approaches on image captioning metrics (Table 2.1). Our performance improves with beam search (Table 2.2).
- Adding attention to our CNN gives improvements on metrics and we outperform the LSTM+Attn baseline (Xu et al., 2015) (Table 2.1). Figure 2.5 shows that with attention we identify salient objects for the given image.
- We analyze the CNN and RNN approaches and show that CNN produces (1) more entropy in the output probability distribution, (2) gives better word prediction accuracy (Figure 2.6), and (3) does not suffer as much from vanishing gradients (Figure 2.8).
- In Table 2.4, we show that a CNN with 1.5× more parameters can be trained in comparable time. This is because we avoid the sequential processing of RNNs.

| Method | Beam Size=2 | | | | | | | | Beam Size=3 | | | | | | | | Beam Size=4 | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | B1 | B2 | B3 | B4 | M | R | C | S | B1 | B2 | B3 | B4 | M | R | C | S | B1 | B2 | B3 | B4 | M | R | C | S |
| LSTM | .715 | .545 | .407 | .304 | .248 | .526 | .940 | .178 | .715 | .544 | .409 | .310 | .249 | .528 | .946 | .178 | .714 | .543 | .410 | .311 | **.250** | .529 | .951 | **.179** |
| CNN | .712 | .541 | .404 | .303 | .248 | .527 | .937 | .178 | .709 | .538 | .403 | .303 | .247 | .525 | .929 | .176 | .706 | .533 | .400 | .302 | .247 | .522 | .925 | .175 |
| CNN+Attn | .718 | .549 | .411 | .306 | .248 | .528 | .942 | .177 | **.722** | **.553** | **.418** | **.316** | **.250** | **.531** | **.952** | **.179** | .718 | .550 | .415 | .314 | .249 | .528 | .951 | .179 |

Table 2.2: Comparison of different methods (metrics same as Table 2.1) with beam search on the output word probabilities. Our results show that with beam size= 3 our CNN outperforms LSTM (Karpathy & Fei-Fei, 2015) on all metrics. Note, compared to Table 2.1, the performance improves with beam search. We use the MS COCO test split for this experiment. For beam search, we pick one caption with maximum log probability (sum of log probability of words) from the top-$k$ beams and report the above metrics for it. Beam = 1 is same as the test set results reported in Table 2.1.

The details of our experimental setup and these results are discussed below. The PyTorch implementation of our convolutional image captioning is available on github.[1]

| | c5 (Beam = 1) | | | | | | | c40 (Beam = 1) | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | B1 | B2 | B3 | B4 | M | R | C | B1 | B2 | B3 | B4 | M | R | C |
| LSTM | .704 | .528 | .384 | .278 | **.241** | **.517** | **.876** | .880 | .778 | .656 | .537 | **.321** | .655 | **.898** |
| CNN+Attn | **.708** | **.534** | **.389** | **.280** | **.241** | **.517** | .872 | **.883** | **.786** | **.667** | **.545** | **.321** | **.657** | .893 |
| | c5 (Beam = 3) | | | | | | | c40 (Beam = 3) | | | | | | |
| | B1 | B2 | B3 | B4 | M | R | C | B1 | B2 | B3 | B4 | M | R | C |
| LSTM | .710 | .537 | .399 | .299 | **.246** | .523 | .904 | .889 | .794 | .681 | .570 | **.334** | .671 | .912 |
| CNN+Attn | **.715** | **.545** | **.408** | **.304** | **.246** | **.525** | **.910** | **.896** | **.805** | **.694** | **.582** | .333 | **.673** | **.914** |

Table 2.3: Above, we show that CNN outperforms LSTM on BLEU metrics and gives comparable scores to LSTM on other metrics for test split on MSCOCO evaluation server. Note, this hidden test split of 40,775 images on the evaluation server is different from the 5000 images test split used in Tables 2.1 and 2.2. We compare our CNN+Attn method to the LSTM baseline (metrics same as Table 2.1). The $c5, c40$ scores above are computed with $5, 40$ reference captions per test image respectively. We show comparison results for beam size 1 and beam size 3 for both the methods.

### 2.6.1 Dataset and Baselines

We conducted experiments on the MS COCO dataset (Lin et al., 2014). Our train/val/test splits follow (Karpathy & Fei-Fei, 2015; Xu et al., 2015). We use 113287 training images, 5000 images for validation, and 5000 for testing. Henceforth, we will refer to our approach as CNN, and our approach with the attention (Section 2.5.1) as CNN+Attn. We use the

[1] https://github.com/aditya12agd5/convcap

following naming convention for our baselines: (Karpathy & Fei-Fei, 2015) is denoted by LSTM and (Xu et al., 2015) is referred to as LSTM+Attn.

### 2.6.2 Comparison on Image Captioning Metrics

We consider multiple conventional evaluation metrics, BLEU-1, BLEU-2, BLEU-3, BLEU-4 (Papineni et al., 2001), METEOR (Denkowski & Lavie, 2014), ROUGE (Lin, 2004), CIDEr (Vedantam et al., 2015) and SPICE (Anderson et al., 2016). See Table 2.1 for the performance on all these metrics for our val/test splits. Note that we obtain comparable CIDEr scores and better SPICE scores than LSTM on test set with our CNN+Attn method. Our BLEU, METEOR, ROUGE scores are less than the LSTM ones, but the margin is very small. Our CNN+Attn method outperforms the LSTM+Attn baseline on the test set for all metrics reported in (Xu et al., 2015). For Table 2.1, we form the caption by choosing the word with maximum probability at each step. The metrics are reported for this one caption formed by choosing the maximum probability word at every step.

Instead of sampling the maximum probability words, we also perform beam search with different beam sizes. We perform beam search for both LSTM and our CNN methods. With beam search, we pick the maximum probability caption (sum of log word probability in the beam). The results reported in Table 2.2 demonstrate that with beam size of 3 we achieve better BLEU, ROUGE, CIDEr scores than LSTM and equal METEOR and SPICE scores.

In Table 2.3, we show the results obtained on the MSCOCO evaluation server. These results are computed over a test set of $40,775$ images for which ground-truth is not publicly available. We demonstrate that our method does better on all BLEU metrics, especially with beam size 3, we perform better than the LSTM based method.

**Comparison to recent state-of-the-art:** For better performance on the MSCOCO leader board we use ResNet features instead of VGG-16. Table 2.5 shows ResNet boosts our performance on the MSCOCO split (cf. Table 2.1) and we compare it to more recent methods (Anderson et al., 2018) and (Yao et al., 2017a). We are almost as good as (Yao et al., 2017a). If we had access to their pre-trained attribute network, we may outperform it. (Anderson et al., 2018) uses a sophisticated attention mechanism, which can be incorporated into our architecture as part of future work.

### 2.6.3 Qualitative Comparison

See Figure 2.4 for a qualitative comparison of captions generated by CNN and LSTM. In Figure 2.5, we overlay the attention parameters on the image for each word prediction.

**LSTM:** a man and a woman in a suit and tie
**CNN:** a black and white photo of a man and woman in a suit
**GT:** A man sitting next to a woman while wearing a suit.

**LSTM:** a cat is laying down on a bed
**CNN:** a polar bear is drinking water from a white bowl
**GT:** A white polar bear laying on top of a pool of water

**LSTM:** a bear is standing on a rock in a zoo
**CNN:** two bears are walking on a rock in the zoo
**GT:** two bears touching noses standing on rocks

**LSTM:** a box of donuts with a variety of toppings
**CNN:** a box of doughnuts with sprinkles and a sign
**GT:**A bunch of doughnuts with sprinkles on them

**LSTM:** a dog and a dog in a field
**CNN:** two cows are standing in a field of grass
**GT:** A dog and a horse standing near each other

Figure 2.4: Captions generated by our CNN are compared to the LSTM and ground-truth caption. In the examples above our CNN can describe things like black and white photo, polar bear/white bowl, number of bears, sign in the donut image which LSTM fails to do. The last image (rightmost) shows a failure case for CNN. Typically we observe that CNN and LSTM captions are of similar quality. We use our CNN+Attn method (Section 2.5.1) and the MSCOCO test split for these results.

**CNN:** a plate of food with broccoli and rice
**GT:** A BBQ steak on a plate next to mashed potatoes and mixed vegetables.

a plate of food with broccoli ... rice

**CNN:** a man sitting on a bench overlooking the ocean
**GT:** A man sitting on top of a bench near the ocean

a man sitting on a bench ... ocean

Figure 2.5: Attention parameters are overlayed on the image. These results show that we focus on salient regions as broccoli, bench when predicting these words and that the attention is uniform when predicting words such as a, of and on.

| Method | # Parameters | Train time per epoch |
|---|---|---|
| LSTM (Karpathy & Fei-Fei, 2015) | 13M | 1529s |
| Our CNN | 19M | 1585s |
| Our CNN+Attn | 20M | 1620s |

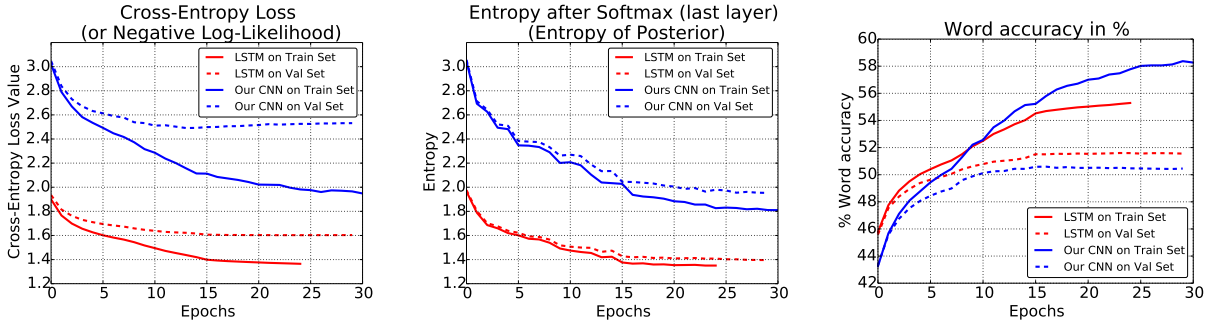
Table 2.4: We train a CNN faster per parameter than the LSTM. This is because CNN is not sequential like the LSTM. We use PyTorch implementation of (Karpathy & Fei-Fei, 2015) and our CNN-based method, and the timings are obtained on Nvidia Titan X GPU.

Note that our attention parameters are $7 \times 7$ as described in Section 2.5.1 and therefore the image is divided in a $7 \times 7$ grid. These results show that our attention focuses on salient objects such as man, broccoli, ocean, bench, *etc.*, when predicting these respective words.

| Method | B1 | B2 | B3 | B4 | M | R | C |
|---|---|---|---|---|---|---|---|
| Our Resnet-101 | .72 | .549 | .403 | .293 | .248 | .527 | .945 |
| Our Resnet-152 | **.725** | **.555** | **.41** | **.299** | **.251** | **.532** | **.972** |
| LSTM Resnet-152 | .724 | .552 | .405 | .294 | **.251** | **.532** | .961 |
| Resnet-152 (Yao et al., 2017a) | .731 | .564 | .426 | .321 | .252 | .537 | .984 |
| Resnet-101 (Anderson et al., 2018) | .772 | - | - | .362 | .27 | .564 | 1.13 |

Table 2.5: Comparison to recent state-of-the-art with Resnet.

Our results also show that the attention is uniform when predicting words such as a, of, on, *etc*., which are unrelated to the image content.




(a) CNN gives higher cross-entropy loss on train/val set of MSCOCO compared to LSTM. But, as we show in (c), CNN obtains better % word accuracy than LSTM. Therefore, it assigns max. probability to correct word. The CNN loss is high because its output probability distributions have more entropy than LSTM.

(b) The entropy of the softmax layer (or posterior probability distribution) of our CNN is higher than the LSTM. For ambiguous problems such as image captioning, it is desirable to have a less peaky (multi-modal) posterior (like ours) capable of producing multiple captions, rather than a peaky one (like LSTM).

(c) Even though the CNN training loss is higher than LSTM, its word prediction accuracy is better than LSTM on train set. On val set, the difference in accuracy between LSTM and CNN is small (only $\sim 1\%$).

Figure 2.6: In the figures above we plot (a) Cross-entropy loss, (b) Entropy of the softmax layer, (c) Word accuracy on train/val set. Blue line denotes our CNN and red denotes the LSTM based method (Karpathy & Fei-Fei, 2015). Solid/dotted lines denote train/val set of MSCOCO respectively. For train set, we randomly sample 10k images and use the entire val set.

### 2.6.4 Analysis of CNN and RNN

In Table 2.4 we report the number of trainable parameters and the training time per epoch. CNNs with $\sim 1.5\times$ parameters can be trained in comparable time.

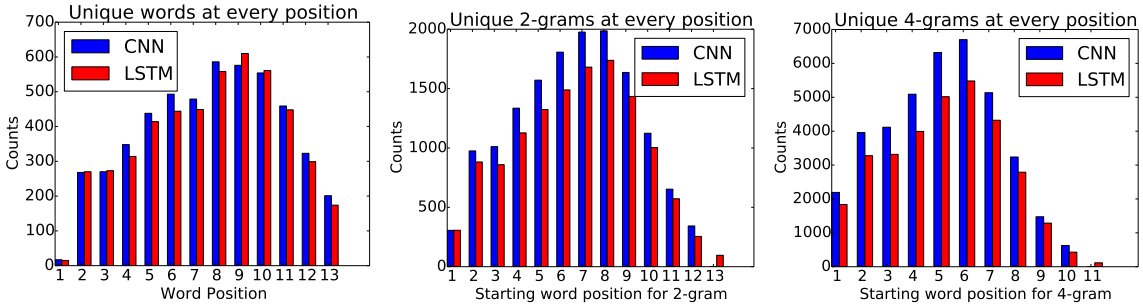
Table 2.1, 2.2 and 2.3 show that we obtain comparable performance from both CNN and RNN/LSTM-based methods. Encouraged by this result, we analyze the characteristics of these two methods. For fair comparison, we use our CNN without attention, since the RNN method does not use spatial image features. First, we compare the negative log-likelihoods (or cross-entropy loss) on a subset of train and the entire val set (see Figure 2.6 (a)). We find that the loss is higher for CNN than RNN. This is because CNNs are being penalized for producing less-peaky word probability distributions. To evaluate this further, we plot the entropy of the output probability distribution (Figure 2.6 (b)) and the classification accuracy, *i.e.*, the number of times the maximum probability word is the ground truth (Figure 2.6 (c)). These plots show that RNNs are good at producing low entropy and therefore peaky word probability distributions at the output, while CNNs produce less peaky distributions (and high entropy). Less peaky distributions are not necessarily bad, particularly for a problem like image captioning, where multiple word predictions are possible. Despite, less peaky distributions, Figure 2.6 (c) shows that the maximum probability word is correct more often on the train set and it is within approx. 1% accuracy on the val set. Note, cross-entropy loss is a proxy for the classification accuracy and we show that CNNs have higher cross entropy loss, but their classification accuracy is good. Less peaky posterior distributions provided by a CNN may be indicative of CNNs being more capable of predicting diverse captions.

**Diversity:** In Figure 2.7, we plot the unique words and 2/4-grams predicted at every word position or time-step. The plot is for word positions 1 to 13. This plot shows that for the CNN we have higher unique words for more word positions and consistently higher 2/4-grams than LSTM. This supports our analysis that CNNs have less peaky (or one-hot) posteriors and therefore can produce more diversity. For these diversity experiments, we perform a beam search with beam size 10 and use all the top 10 beams.

**Vanishing Gradient:** Since RNNs/LSTMs are known to suffer from vanishing gradient problems, in Figure 2.8, we plot the gradient norm at the output embedding/classification layer and the gradient norm at the input embedding layer. The values are averaged over 1 training epoch. These plots show that the gradients in RNN/LSTM diminishes more than the ones in CNNs. Hence RNN/LSTM nets are more likely to suffer from vanishing gradients, which stalls learning. If learning is stalled, for larger datasets than the ones we currently use for image captioning, the performance of RNN and CNN may differ significantly.

## 2.7 RELATED WORK

Describing the content of an observed image is related to a large variety of tasks. Object detection (Redmon et al., 2015; Ren et al., 2015; Yeh et al., 2017) and semantic segmenta-




(a) Unique words (b) Unique 2-grams (c) Unique 4-grams

Figure 2.7: We perform beam search of beam size 10 with our best performing LSTM and CNN models. We use the top 10 beams to plot the unique words, 2/4-grams predicted for every word position. CNN (blue) produces higher unique words, 2/4-grams at more positions, and therefore more diversity, than LSTM (red).

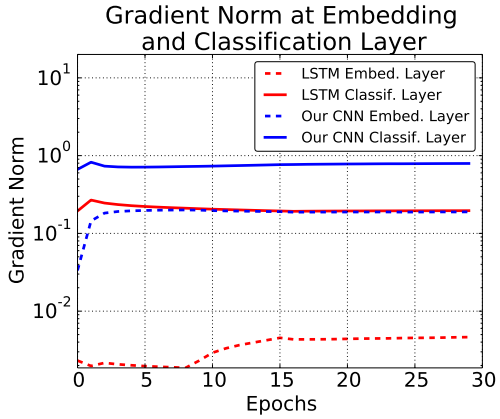



Figure 2.8: Here, we plot the gradient norm at the input embedding (dotted line) and output embedding/classification (solid line) layer. The gradient to the first layer of LSTM decays by a factor $\sim 100$ in contrast to our CNN, where it decays by a factor of $\sim 10$. There is prior evidence in literature that unlike CNNs, RNN/LSTMs suffer from vanishing gradients (Pascanu et al., 2013; Sutskever et al., 2011).

tion (Mostajabi et al., 2015; Shelhamer et al., 2017; Hu et al., 2017) can be used to obtain a list of objects. Detection of co-occurrence patterns and relationships between objects can help to form sentences. Generating sentences by taking advantage of surrogate tasks is then a multi-step approach which is beneficial for interpretability but lacks a joint objective that can be trained end-to-end.

Early techniques formulate image captioning as a retrieval problem and find the best fitting description from a pool of possible captions (Hodosh et al., 2013; Jia et al., 2011; Ordonez et al., 2011; Socher et al., 2014b). Those techniques are built upon the idea that the fitness between available textual descriptions and images can be learned. While this permits

end-to-end training, matching image descriptors to a sufficiently large pool of captions is computationally expensive. In addition, constructing a database of captions that is sufficient for describing a reasonably large fraction of images seems prohibitive.

To address this issue, recurrent neural nets (RNNs) or probabilistic models like Markov chains, which decompose the space of a caption into a product space of individual words are compelling. The success of RNNs for image captioning is based on a key component, *i.e.*, the Long-Short-Term-Memory (LSTM) (Hochreiter & Schmidhuber, 1997b) or recent alternatives like the gated recurrent unit (GRU) (Cho et al., 2014). These components capture long-term dependencies by adding a memory cell, and they address the vanishing or exploding gradient issue of classical RNNs to some degree.

Based on this success, Mao et al. (2015) train a vision (or image) CNN and a language RNN that shares a joint embedding layer. Vinyals et al. (2015a) jointly train a vision (or image) CNN with a language RNN to generate sentences, Xu et al. (2015) extends Vinyals et al. (2015a) with additional attention parameters and learns to identify salient objects for caption generation. Karpathy & Fei-Fei (2015) use a bi-directional RNN along with a structured loss function in a shared vision-language space. Yao et al. (2017a) use an additional network trained on coco-attributes, and Anderson et al. (2018); Schwartz et al. (2017) develop an attention mechanism for captioning. These recurrent neural nets have found widespread use for captioning because they have been shown to produce remarkably fitting descriptions.

Despite the fact that the above RNNs based on LSTM/GRU deliver remarkable results, *e.g.*, for image captioning, their training procedure is all but trivial. For instance, while the forward pass during training can be in parallel across samples, it is inherently sequential in time, limiting the parallelism. To address this issue, van den Oord et al. (2016) proposed a PixelCNN architecture for conditional image generation that approximates an RNN. Gehring et al. (2017) and Vaswani et al. (2017) demonstrate that convolutional architectures with attention achieve state-of-the-art performance on machine translation tasks. In spirit similar is our approach for image captioning, which is convolutional but addresses a different task.

## 2.8 CONCLUSION

We discussed a convolutional approach for image captioning and showed that it performs on par with existing LSTM techniques. We also analyzed the differences between RNN based learning and our method, and found gradients of lower magnitude as well as overly confident predictions to be existing LSTM network concerns.

# Part II

# Learning Models that Generate Diverse Outputs in a Computationally Efficient Manner

# CHAPTER 3: FAST, DIVERSE AND ACCURATE IMAGE CAPTIONING GUIDED BY PART-OF-SPEECH

## 3.1 INTRODUCTION

In this chapter we show how to encourage an image captioning system to generate diverse captions by conditioning on different high-level summaries of the image. Our summaries are quantized part-of-speech (POS) tag sequences. Our system generates captions by (a) predicting different summaries from the image then (b) predicting captions conditioned on each summary. This approach leads to captions that are *accurate, quick to obtain*, and *diverse*. Our system is accurate, because it is able to steer a number of narrow beam searches to explore the space of caption sequences more efficiently. It is fast because each beam is narrow. And the captions are diverse, because depending on the summary (*i.e.*, part-of-speech) the system is forced to produce captions that contain (for example) more or less adjectives. This means we can avoid the tendency to produce minimal or generic captions that is common in systems that try to optimize likelihood without awareness of language priors (like part-of-speech).

A large body of literature has focused on developing predictive image captioning techniques, often using recurrent neural nets (RNN) (Mao et al., 2015; Vinyals et al., 2015b; Xu et al., 2015; Karpathy & Fei-Fei, 2015; Anderson et al., 2018). More recently Aneja et al. (2018); Wang & Chan (2018), demonstrate predictive captioning with accuracy similar to RNNs while using convolutional networks. An essential feature of captioning is that it is ambiguous – many captions can describe the same image. This creates a problem, as image captioning programs trained to maximize some score may do so by producing strongly non-committal captions. It also creates an opportunity for research – how can one produce multiple, diverse captions that still properly describe the image? Our method offers a procedure to do so.

Tractable image captioning involves factoring the sequence model for the caption. Inference then requires beam search, which investigates a set of captions determined by local criteria to find the caption with highest posterior probability. Finding very good captions requires a wide beam, which is slow. Moreover, beam search is also known to generate generic captions that lack diversity (Finkel et al., 2006; Gimpel et al., 2013). Variational auto-encoder (VAE) (Wang et al., 2017c) and generative adversarial net (GAN) (Dai et al., 2017; Shetty et al., 2017; Li et al., 2018) formulations outperform beam search on diversity metrics. VAE and GAN-based methods sample latent vectors from some distribution, then generate captions conditioned on these samples. The latent variables have no exposed se-

| Method | Fast | Diverse | Accurate |
|---|---|---|---|
| Beam search | × | × | ✓ |
| Diverse beam search (Vijayakumar et al., 2018) | × | × | ✓ |
| AG-CVAE (Wang et al., 2017c) | ✓ | ✓ | × |
| Ours (POS) | ✓ | ✓ | ✓ |

Table 3.1: We show that our part-of-speech (POS) based method achieves the trifecta of **high accuracy, fast computation** and **more diversity**. Beam search and diverse beam search are slow. They also produce captions with high mutual overlap and lower distinct $n$-grams than POS (see mBleu-4, div-1 and div-2 in Table 4.2). POS and AG-CVAE are fast, however POS does better on captioning metrics in Figure 3.3 and is therefore more accurate.

mantics, and captions tend not to score as well (on captioning metrics) as those produced by beam search (*e.g.*, Tab. 1 of Shetty et al. (2017)).

This chapter offers an alternative. First predict a meaningful summary of the image, then generate the caption based on that summary. For this to work, the summary needs to be able to drive language generation (for the caption generator), and must be predictable. We find quantized part of speech tag sequences to be very effective summaries. These sequences can clearly drive language generation (*e.g.*, forcing a captioner to produce adjectives in particular locations). More surprisingly, one can predict quantized tag sequences from images rather well, likely because such sequences do summarize the main action of the image. For example, compare *determiner-noun-verb* with *determiner-adjective-noun-verb-adjective-noun*. In the first case, something appears in the image, in the second, a subject with a noteworthy attribute is doing something to an object with a noteworthy attribute. Consequently, the two images appear quite different.

**Contributions:** We show that image captioning with POS tag sequences is fast, diverse and accurate (Table 3.1). Our POS methods sample captions faster and with more diversity than techniques based on beam search and its variant diverse beam search (Vijayakumar et al., 2018) (Table 4.2). Our diverse captions are more accurate than their counterparts produced by GANs (Shetty et al., 2017) (Table 3.5) and VAEs (Wang et al., 2017c) (Table 3.3, Figure 3.3).

## 3.2 BACKGROUND

**Problem Setup and Notation:** The goal of diverse captioning is to generate $k$ sequences $y^1, y^2, \ldots, y^k$, given an image. For readability we drop the super-script and focus on a single

sequence $y$. The methods we discuss and develop will sample many such sequences $y$ and rank them to obtain the best-$k$ – $y^1, y^2, \ldots, y^k$. A single caption $y = (y_1, \ldots, y_N)$ consists of a sequence of words $y_i$, $i \in \{1, \ldots, N\}$ which accurately describe the given image $I$. For each caption $y$, the words $y_i, i \in \{1, \ldots, N\}$ are obtained from a fixed vocabulary $\mathcal{Y}$, *i.e.*, $y_i \in \mathcal{Y}$. Additionally, we assume availability of a part-of-speech (POS) tagger for the sentence $y$. More specifically, the POS tagger provides a tag sequence $t = (t_1, \ldots, t_N)$ for a given sentence, where $t_i \in \mathcal{T}$ is the POS tag for word $y_i$. The set $\mathcal{T}$ encompasses 12 universal POS tags – *verb (VERB), noun (NOUN), pronoun (PRON), etc.*[1]

To train our models we use a dataset $\mathcal{D} = \{(I, y, t)\}$ which contains tuples $(I, y, t)$ composed of an image $I$, a sentence $y$, and the corresponding POS tag sequence $t$. Since it is not feasible to annotate the $\sim$ .5M captions of MSCOCO with POS tags, we use an automatic part-of-speech tagger.[1]

**Classical Image Captioning:** Classical techniques factor the joint probabilistic model $p_\theta(y|I)$ over all words into a product of conditionals. They learn model parameters $\theta^*$ by maximizing the likelihood over the training set $\mathcal{D}$, *i.e.*,

$$\max_{\theta} \sum_{(I,y)\in\mathcal{D}} \log p_\theta(y|I), \quad \text{where} \quad p_\theta(y|I) = \prod_{i=1}^{N} p(y_i|y_{<i}, I). \tag{3.1}$$

The factorization of the joint probability distribution enforces a temporal ordering of words. Hence, word $y_i$ at the $i^{\text{th}}$ time-step (or word position) depends only on all previous words $y_{<i}$. This probability model is represented using a recurrent neural network or a feed-forward network with temporal (or masked) convolutions. Particularly the latter, *i.e.*, temporal convolutions, have been used recently for different vision and language tasks in place of classical recurrent neural nets (Aneja et al., 2018; Gehring et al., 2017; Bai et al., 2018).

During training, we learn the optimal parameters $\theta^*$. Then for test image $I$, conditional word-wise posterior probabilities $p_{\theta^*}(y_i|y_{<i}, I)$ are generated sequentially from $i = 1$ to $N$. Given these posteriors, beam search is applicable and forms our baseline. Figure 3.1 illustrates beam search with a beam width of $k$ from word position $y_i$ to $y_{i+1}$. Here, beam search maintains best-$k$ (incomplete) captions ordered by likelihood. It expands the best-$k$ captions at every word greedily from start to end of the sentence.

More specifically, for beam search from word position $i$, we first generate posteriors $p^j_{\theta^*}(y_{i+1}|y^j_{<(i+1)}, I)$ based on the current top-$k$ list containing $y^j_{<(i+1)}$, $j \in \{1, \ldots, k\}$. We then obtain new top-$k$ captions by expanding each of the $k$ entries $y^j_{<(i+1)}$ in the list using the computed posterior $p^j_{\theta^*}(y_{i+1}|y^j_{<(i+1)}, I)$. We call this 'expand top-$k$.' The time complexity

[1] See https://www.nltk.org/book/ch05.html for POS tag and automatic POS tagger details

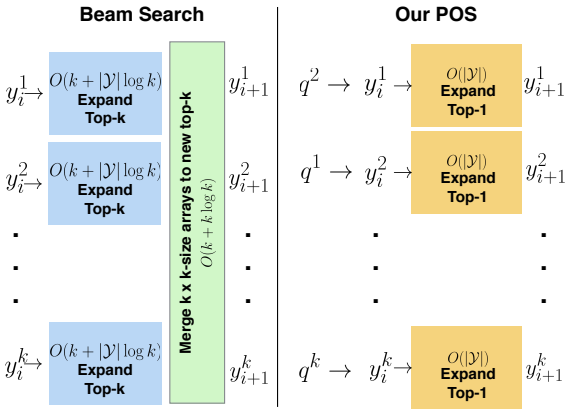



Figure 3.1: Illustration of beam search and POS-sampling to expand the best-$k$ captions $(y_i^1, y_i^2, \dots y_i^k)$ from word position $i$ to $i+1$. See Section 6.2 for notation and other details.

for a single expand top-$k$ operation is identical to obtaining the sorted top-$k$ values from an array of size $|\mathcal{Y}|$.[2] The time complexity of all expand top-$k$ operations is $O(k^2 + |\mathcal{Y}|k \log k)$.

We merge all the expanded top-$k$ captions to the final top-$k$ captions using the log sum of the posterior probability at word position $i+1$. We call this operation merge. The merge operation has a complexity of $O(k + k \log k)$, which is identical to merging $k$ sorted arrays.[3] In Section 3.3, we show that our inference with POS has better time complexity.

### 3.3 IMAGE CAPTIONING WITH PART-OF-SPEECH

In our approach for image captioning, we introduce a POS tag sequence $t$, to condition the recurrent model given in Eq. (3.1). More formally, we use the distribution

$$p_\theta(y|t, I) = \prod_{i=1}^{N} p_\theta(y_i|t, y_{<i}, I). \tag{3.2}$$

Following classical techniques, we train our POS-conditioned approach by maximizing the likelihood (similar to Eq. (3.1)), *i.e.*, we want to find the parameters

$$\theta^* = \arg\max_\theta \sum_{(I,t,y)\in\mathcal{D}} \log p_\theta(y|t, I). \tag{3.3}$$

Importantly, note that we use the entire POS tag sequence in the conditional above, because it allows global control over the entire sentence structure.

[2] https://www.geeksforgeeks.org/k-largestor-smallest-elements-in-an-array/
[3] https://www.geeksforgeeks.org/merge-k-sorted-arrays/

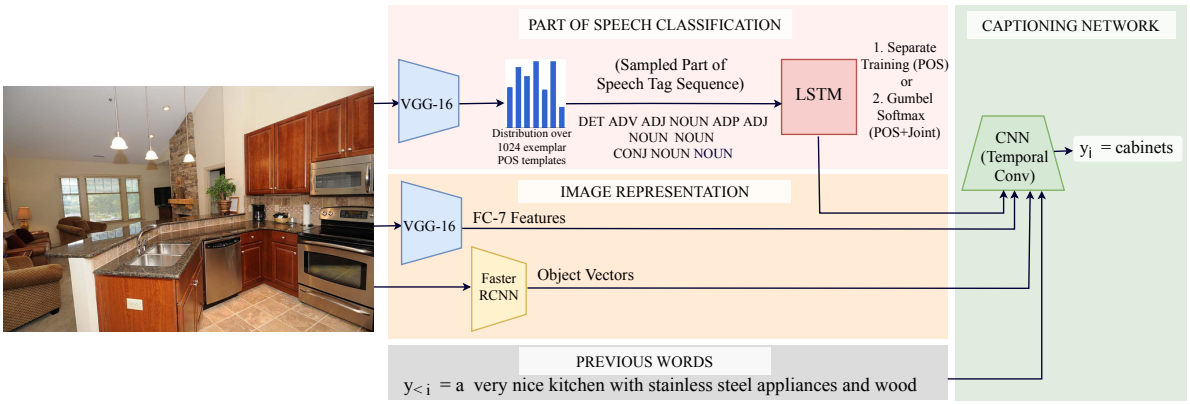



Figure 3.2: An illustration of our POS captioning method on a test image. For the image representation, fc7 features are extracted from VGG-16 and embedded into 512 dimensional vectors. For object vectors, we use the 80 dimensional class vector from faster rcnn (Ren et al., 2015) (same as Wang et al. (2017c)). For part-of-speech classification, we use VGG-16 with two linear layers and a 1024-way softmax. Then, we encode sampled POS via an LSTM-net to a 512 dimensional vector. Our captioning network uses temporal convolutions and operates on image representation, part-of-speech vector, object vector and previous words in the caption ($y_{<i}$) to produce the next word ($y_i$). The network is trained for 20 epochs using the ADAM optimizer (Kingma & Ba, 2015) (initial learning rate of $5e^{-5}$ and a decay factor of .1 after 15 epochs). The part of speech classification step can be trained separately (POS) or jointly using a gumbel softmax (POS+Joint). Note, image representation is same for our method and baselines.

Training involves learning the parameters $\theta^*$ for our conditional captioning model (Eq. (3.3)). During test time, conditioning on POS tags provides a mechanism for diverse image captioning, *i.e.*, given a test image $I$, we obtain $k$ diverse captions by sampling $k$ POS tag sequences $t^1, t^2, \ldots, t^k$. Note that every sequence is a tuple of POS tags, *i.e.*, $t^i = (t^i_1, t^i_2, \ldots)$, $i \in \{1, \ldots, k\}$.

Since a large number of possible POS tag sequences exists, in Section 3.3.1, we discuss how we obtain quantized POS tag sequences $q^1, q^2, \ldots, q^k$ given the input image. These quantized sequences approximate the actual POS tag sequences $t^1, t^2, \ldots, t^k$.

Concretely, during inference we sample $k$ quantized POS tag sequences given the image. This is shown as the part-of-speech classification step in Figure 3.2. Then, we encode each sampled POS tag sequence $q$ using an LSTM model. The encoded POS tag sequence, along with object vector, image features (fc7 of VGG-16) and previous words ($y_{<i}$) forms the input to the temporal convolutions-based captioning network. This captioning network implements our posterior probability $p_\theta(y_i|y_{<i}, q, I)$, which is used to predict the next word $y^*_i = \arg\max_{y_i} p_\theta(y_i|y_{<i}, q, I)$.

**Fast inference with POS:** For every sampled tag sequence $q^j, j \in \{1, 2, \cdots, k\}$ (*i.e.*

quantization of tag sequence $t^j$), we maximize the learned probabilistic model, *i.e.*, $y_i^j = \arg\max_y p_{\theta^*}(y_i|y_{<i}, q^j, I)$ greedily. As just discussed, we simply use the maximum probability word at every word position. Figure 3.1 compares this computationally much more effective method, which has a time complexity of $O(k|\mathcal{Y}|)$, to the breadth first approach employed by beam search.

Note that POS-based sampling requires only a single max-operation at every step during inference (our effective beam size is 1), making it faster than beam search with wide beams. It is also faster than diverse beam search (with group size parameter set to 1 as in our results) which performs the $k$ 'expand top-$k$' operations sequentially using an augmented diversity function.

### 3.3.1 Image to Part-of-Speech Classification

Because our model conditions sentence probabilities on a POS tag sequence, we need to compute it before performing inference. Several ways exist to obtain the POS tag sequence. *E.g.*, choosing a POS tag sequence by hand, sampling from a distribution of POS tag sequences seen in the dataset $\mathcal{D}$, or predicting POS tag sequences conditioned on the observed image $I$. The first one is not scalable. The second approach of sampling from $\mathcal{D}$ without considering the provided image is easy, but generates inaccurate captions. We found the third approach to yield most accurate results. While this seems like an odd task at first, our experiments suggest very strongly that image based prediction of POS tag sequences works rather well. Indeed, intuitively, inferring a POS tag sequence from an image is similar to predicting a situation template (Yatskar et al., 2016) – one must predict a rough template sketching what is worth to be said about an image.

To capture multi-modality, we use a classification model to compute our POS predictions for a given image $I$. However, we find that there are $> 210K$ POS tag sequences in our training dataset $\mathcal{D}$ of $|\mathcal{D}| > 500K$ captions. To maintain efficiency, we therefore quantize the space of POS tag sequences to 1024 exemplars as discussed subsequently.

**Quantizing POS tag sequences:** We perform a hamming distance based k-medoids clustering to obtain 1024-cluster centers. We use concatenated 1-hot encodings (of POS tags) to encode the POS tag sequence. We observe our clusters to be tight, *i.e.*, more than 75% of the clusters have an average hamming distance less than 3. We use the cluster medoids as the quantized POS tag sequences for our classifier. Given an input tag sequence $t$ we represent it using its nearest neighbor in quantized space, which we denote by $q = \mathcal{Q}(t)$. Note, in our notation the quantization function $\mathcal{Q}(t)$, reduces $t$ to its quantized tag sequence $q$.

Our image to part-of-speech classifier (shown in Figure 3.2) learns to predict over quantized POS sequence space by maximizing the likelihood, $p_\phi(q|I)$. Formally, we look for its optimal parameters $\phi^*$ via

$$\phi^* = \arg\max_{\phi} \sum_{(I,t)\in\mathcal{D}} \log p_\phi(q|I), \tag{3.4}$$

where $\log p_\phi(q|I) = \sum_{i=1}^{1024} \delta[q^i = \mathcal{Q}(t)] \log p_\phi(q^i|I)$.

### 3.3.2 Separate vs. Joint Training

Training involves learning the parameters $\theta$ of the captioning network (Eq. (3.3)) and the parameters $\phi$ of the POS classification network (Eq. (3.4)). We can trivially train these two networks separately and we call this method **POS**.

We also experiment with joint training by sampling from the predicted POS posterior $p_\phi(t|I)$ using a Gumbel soft-max (Jang et al., 2017) before subsequently using its output in the captioning network. Inconsistencies between sampled POS sequence and corresponding caption $y$ will introduce noise since the ground-truth caption $y$ may be incompatible with the sampled sequence $q$. Therefore, during every training iteration, we sample 50 POS tag sequences from the Gumbel soft-max and only pick the one $q$ with the best alignment to POS tagging of caption $y$. We refer to this form of joint training via **POS+Joint**. In Section 4.3.1 and Section 3.4.2, we show that POS+Joint (*i.e.*, jointly learning $\theta$ and $\phi$) is useful and produces more accurate captions.

## 3.4 RESULTS

In the following, we compare our developed approach for diverse captioning with POS tags to competing baselines for diverse captioning. We first provide information about the dataset, the baselines and the evaluation metrics before presenting our results.

**Dataset:** We use the **MS COCO** dataset (Lin et al., 2014) for our experiments. For the train/val/test splits we follow: (1) M-RNN (Mao et al., 2015) using 118,287 images for training, 4,000 images for validation, and 1,000 images for testing; and (2) Karpathy & Fei-Fei (2015) using 113,287 images for training, 5,000 images for validation and 5,000 images for testing. The latter split is used to compare to GAN-based results in Table 3.5.

**Methods:** In the results, we denote our approach by **POS**, and our approach with joint training by **POS+Joint** (see Section 3.3.2 for the differences). We compare to the additive Gaussian conditional VAE-based diverse captioning method of Wang et al. (2017c),

| Method | Beam size or #samples | Best-1 Oracle Accuracy | | | | | | | | Speed (s/img) | Speed | Accuracy |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | | B4 | B3 | B2 | B1 | C | R | M | S | | | |
| Beam search | 20 | 0.489✓ | 0.626✓ | 0.752✓ | 0.875✓ | 1.595✓ | 0.698✓ | 0.402✓ | 0.284✓ | 3.74× | × | ✓ |
| Div-BS | | 0.383× | 0.538× | 0.687× | 0.837 | 1.405 | 0.653 | 0.357 | 0.269 | 3.42 | × | × |
| AG-CVAE | | 0.471 | 0.573 | 0.698 | 0.834× | 1.308× | 0.638× | 0.309× | 0.244× | - | - | × |
| POS | | 0.449 | 0.593 | 0.737 | 0.874 | 1.468 | 0.678 | 0.365 | 0.277 | 0.21 | ✓ | ✓ |
| POS+Joint | | 0.431 | 0.581 | 0.721 | 0.865 | 1.448 | 0.670 | 0.357 | 0.271 | 0.20✓ | ✓ | ✓ |
| Beam Search | 100 | 0.641✓ | 0.742✓ | 0.835✓ | 0.931✓ | 1.904✓ | 0.772✓ | 0.482✓ | 0.332✓ | 20.33 | × | ✓ |
| Div-BS | | 0.402× | 0.555× | 0.698× | 0.846× | 1.448× | 0.666× | 0.372 | 0.290 | 19.05 | × | × |
| AG-CVAE | | 0.557 | 0.654 | 0.767 | 0.883 | 1.517 | 0.690 | 0.345× | 0.277× | - | - | × |
| POS | | 0.578 | 0.689 | 0.802 | 0.921 | 1.710 | 0.739 | 0.423 | 0.322 | 1.29 | ✓ | ✓ |
| POS+Joint | | 0.550 | 0.672 | 0.787 | 0.909 | 1.661 | 0.725 | 0.409 | 0.311 | 1.27✓ | ✓ | ✓ |

Table 3.2: **Best-1 accuracy by oracle re-ranking**. Our POS methods are faster at sampling than beam search and they also generate a higher scoring best-1 caption than AG-CVAE (Wang et al., 2017c) and Div-BS (Vijayakumar et al., 2018). Beam search obtains the best scores, however it is slow. From all sampled captions (#samples = 20 or 100), we use oracle to pick the best-1 caption for every metric. This gives an estimate of the upper bound on captioning accuracy for each method. We use standard captioning metrics, BLEU (B1-B4) (Papineni et al., 2001), CIDEr (C) (Vedantam et al., 2015), ROUGE (R) , METEOR (M) (Denkowski & Lavie, 2014) and SPICE (S) (Anderson et al., 2016). Note, ✓indicates good performance on the metric for the corresponding column and × indicates bad performance.

denoted by **AG-CVAE**. Our captioning network is based on Aneja et al. (2018). For a fair comparison to beam search we also compare to convolutional captioning (Aneja et al., 2018) with beam search. This is referred to as **beam search**. We compare to diverse beam search denoted denoted by **Div-BS**. The abbreviation **GAN** is used to denote the GAN-based method by Shetty et al. (2017).

**Evaluation criteria:** We compare all methods using four criteria – accuracy, diversity, speed, human perception:

- **Accuracy.** In Section 4.3.1 (Best-1 Accuracy) we compare the accuracy using the standard image captioning task of generating a single caption. Subsequently, in Section 3.4.2 (Best-$k^{\text{th}}$ Accuracy), we assess the accuracy of $k$ captions on different image captioning metrics.
- **Diversity.** We evaluate the performance of each method on different diversity metrics in Section 4.3.2.
- **Speed.** In addition to accuracy, in Section 3.4.4, we also measure the computational efficiency of each method for sampling multiple captions.
- **Human perception.** We perform a user study in Section 3.4.5.

| Method | Beam size or #samples | Best-1 Consensus Re-ranking Accuracy | | | | | | | | Speed (s/img) | Speed | Accuracy |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | | B4 | B3 | B2 | B1 | C | R | M | S | | | |
| Beam search (w. Likelihood) | 20 | 0.305 | 0.402$^{\times}$ | 0.538$^{\times}$ | 0.709$^{\times}$ | 0.947$^{\times}$ | 0.523 | 0.248 | 0.175 | 3.19 | $\times$ | $\times$ |
| Beam search | | 0.319 | 0.423 | 0.564 | 0.733 | 1.018 | 0.537$^{\checkmark}$ | 0.255 | 0.185 | 7.41 | $\times$ | $\checkmark$ |
| Div-BS | | 0.320$^{\checkmark}$ | 0.424$^{\checkmark}$ | 0.562 | 0.729 | 1.032$^{\checkmark}$ | 0.536 | 0.255$^{\checkmark}$ | 0.184 | 7.60$^{\times}$ | $\times$ | $\checkmark$ |
| AG-CVAE | | 0.299$^{\times}$ | 0.402$^{\times}$ | 0.544 | 0.716 | 0.963 | 0.518$^{\times}$ | 0.237$^{\times}$ | 0.173$^{\times}$ | - | - | $\times$ |
| POS | | 0.306 | 0.419 | 0.570$^{\checkmark}$ | 0.744$^{\checkmark}$ | 1.014 | 0.531 | 0.252 | 0.188$^{\checkmark}$ | 1.13$^{\checkmark}$ | $\checkmark$ | $\checkmark$ |
| POS+Joint | | 0.305 | 0.415 | 0.563 | 0.737 | 1.020 | 0.531 | 0.251 | 0.185 | 1.13$^{\checkmark}$ | $\checkmark$ | $\checkmark$ |
| Beam search (w. Likelihood) | 100 | 0.300$^{\times}$ | 0.397$^{\times}$ | 0.532$^{\times}$ | 0.703$^{\times}$ | 0.937$^{\times}$ | 0.519$^{\times}$ | 0.246 | 0.174$^{\times}$ | 18.24 | $\times$ | $\times$ |
| Beam search | | 0.317 | 0.419 | 0.558 | 0.729 | 1.020 | 0.532 | 0.253 | 0.186 | 40.39$^{\times}$ | $\times$ | $\checkmark$ |
| Div-BS | | 0.325$^{\checkmark}$ | 0.430$^{\checkmark}$ | 0.569$^{\checkmark}$ | 0.734 | 1.034 | 0.538$^{\checkmark}$ | 0.255$^{\checkmark}$ | 0.187 | 39.71 | $\times$ | $\checkmark$ |
| AG-CVAE | | 0.311 | 0.417 | 0.559 | 0.732 | 1.001 | 0.528 | 0.245$^{\times}$ | 0.179 | - | - | $\times$ |
| POS | | 0.311 | 0.421 | 0.567 | 0.737 | 1.036 | 0.530 | 0.253 | 0.188$^{\checkmark}$ | 7.54 | $\checkmark$ | $\checkmark$ |
| POS+Joint | | 0.316 | 0.425 | 0.569$^{\checkmark}$ | 0.739$^{\checkmark}$ | 1.045$^{\checkmark}$ | 0.532 | 0.255$^{\checkmark}$ | 0.188$^{\checkmark}$ | 7.32 | $\checkmark$ | $\checkmark$ |

Table 3.3: **Best-1 accuracy by consensus re-ranking**. Our POS methods obtain higher scores on captioning metrics than AG-CVAE (Wang et al., 2017c). This demonstrates our POS natural language prior is more useful than the abstract latent vector used by VAE-based methods. POS methods obtain comparable accuracy to Beam Search and Div-BS (Vijayakumar et al., 2018), and they are more computationally efficient at sampling (*i.e.*, high speed). Note, we also outperform the standard beam search that uses likelihood based ranking. For these results, consensus re-ranking (Devlin et al., 2015) is used to pick the best-1 caption from all sampled captions (unless 'w. Likelihood' is specified). For fair comparison, each method uses the same 80-dimensional object vector from faster rccn (Ren et al., 2015) and the same image features/parameters for consensus re-ranking. The captioning metrics are the same as in Table 3.2. Note, ✓indicates good performance on the metric for the corresponding column and $\times$ indicates bad performance.

### 3.4.1 Best-1 Accuracy

We use two ranking methods – oracle and consensus re-ranking – on the set of generated captions and pick the best-1 caption. Our results for oracle re-ranking in Table 3.2 and for consensus re-ranking in Table 3.3 show that, beam search and diverse beam search are accurate however slow. POS is both fast and accurate. POS outperforms the accuracy of AG-CVAE.

**Oracle re-ranking:** The reference captions of the test set are used and the generated caption with the maximum score for each metric is chosen as best-1 (as also used by Wang et al. (2017c)). This metric permits to assess the best caption for each metric and the score provides an upper bound on the achievable best-1 accuracy. Higher oracle scores are also indicative of the method being a good search method in the space of captions. Results in Table 3.2 show that beam search obtains the best oracle scores. However, it is painfully slow ($\sim$ 20s per image to sample 100 captions). POS, POS+Joint obtain higher accuracy than AG-CVAE and comparable accuracy to beam search with faster runtime.

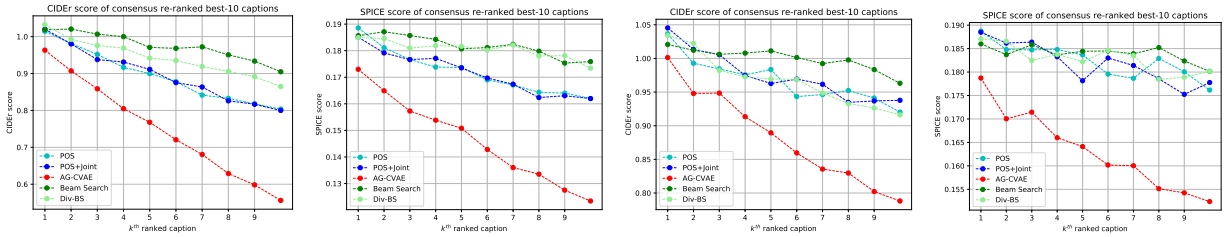
| Method | Beam size or #samples | Distinct Captions | # Novel sentences (Best-5) | mBleu-4 (Best-5) | $n$-gram Diversity (Best-5) Div-1 | Div-2 | Overall Diversity |
|---|---|---|---|---|---|---|---|
| Beam search | 20 | **100%** | 2317 | 0.777 | 0.21 | 0.29 | × |
| Div-BS | | **100%** | 3106 | 0.813 | 0.20 | 0.26 | × |
| AG-CVAE | | 69.8% | 3189 | 0.666 | **0.24** | 0.34 | ✓ |
| POS | | 96.3% | 3394 | **0.639** | **0.24** | **0.35** | ✓ |
| POS+Joint | | 77.9% | **3409** | 0.662 | 0.23 | 0.33 | ✓ |
| Beam search | 100 | **100%** | 2299 | 0.781 | 0.21 | 0.28 | × |
| Div-BS | | **100%** | 3421 | 0.824 | 0.20 | 0.25 | × |
| AG-CVAE | | 47.4% | 3069 | 0.706 | **0.23** | 0.32 | ✓ |
| POS | | 91.5% | **3446** | **0.673** | **0.23** | **0.33** | ✓ |
| POS+Joint | | 58.1% | 3427 | 0.703 | 0.22 | 0.31 | ✓ |
| Human | 5 | 99.8% | - | 0.510 | 0.34 | 0.48 | |

Table 3.4: **Diversity statistics:** For each method, we report the number of novel sentences (*i.e.*, sentences not seen in the training set) out of at most best-5 sentences after consensus re-ranking. Though Beam Search showed high accuracy in Table 3.2, 3.3 and Figure 3.3, here, we see that it produces less number of novel sentences than our POS methods. Therefore, beam search is more prone to regurgitating training data. Low mBleu-4 indicates lower 4-gram overlap between generated captions and more diversity in generated captions. POS has the lowest mBleu-4 and therefore high diversity in generated captions. For details on other metrics see Section 4.3.2.

**Consensus re-ranking scores:** In a practical test setting, reference captions of the test set won't be available to rank the best $k$ captions and obtain best-1. Therefore, in consensus re-ranking, the reference captions of training images similar to the test image are retrieved. The generated captions are ranked via the CIDEr score computed with respect to the retrieved reference set (Devlin et al., 2015).

We use the same image features as Wang et al. (2017b) and parameters for consensus re-ranking as Wang et al. (2017c). Table 3.3 shows that our methods POS and POS+Joint outperform the AG-CVAE baseline on all metrics. Moreover, our methods are faster than beam search and diverse beam search. They produce higher CIDEr, Bleu-1,2, METEOR and SPICE scores. Other scores are comparable and differ in the 3$^{\text{rd}}$ decimal. Note, our POS+Joint achieves better scores than POS, especially for 100 samples. This demonstrates that joint training is useful.

We also train our POS+Joint method on the train/test split of Karpathy & Fei-Fei (2015) used by the GAN method (Shetty et al., 2017). In Table 3.5, we show that we obtain higher METEOR and SPICE scores than those reported in Shetty et al. (2017).




(a) Best-10 CIDEr from 20 samples (b) Best-10 SPICE from 20 samples (c) Best-10 CIDEr from 100 samples (d) Best-10 SPICE from 100 samples

Figure 3.3: **Best-10 CIDEr and SPICE accuracy.** Our POS and POS+Joint achieve best-$k$ accuracy comparable to Beam Search and Div-BS (Vijayakumar et al., 2018) with faster computation time. We outperform the best-$k$ scores of AG-CVAE (Wang et al., 2017c), demonstrating part-of-speech conditioning is better than abstract latent variables of a VAE. Note, this figure is best viewed in high-resolution.

### 3.4.2 Best-$k^{\text{th}}$ Accuracy

Our captioning method can be conditioned on different part-of-speech tags to generate diverse captions. For diverse image captioning, in addition to best-1 accuracy, best-$k^{\text{th}}$ accuracy should also be measured. Best-$k^{\text{th}}$ accuracy is the score of the $k^{\text{th}}$ ranked caption, therefore it is lower than the best-1 score. All $k$ generated captions should be accurate and therefore it is desirable to have high best-$k^{\text{th}}$ scores. This metric has not been reported previously (Shetty et al., 2017; Wang et al., 2017c).

In Figure 3.3, we compare best-$k^{\text{th}}$ ($k = 1$ to 10) scores for all methods. Note, the accuracy of AG-CVAE drops drastically on both CIDEr and Spice, while our POS methods maintain accuracy comparable to beam search. This proves that our POS image summaries are better at sampling accurate captions than the abstract latent variables of a VAE.

### 3.4.3 Evaluation of Diversity

In Table 4.2 we compare methods on diversity metrics.

(1) **Uniqueness:** The number of unique sentences generated after sampling. Beam search and diverse beam search always sample a unique sentence. Note, our POS also samples a high number of unique sentences 19.26 (96.3%) out of 20, 91.55 out of 100. The uniqueness reduces for joint training. This is because, generation of a caption while training POS+Joint is based on a noisy POS tag sequence sampled from the Gumbel softmax. Therefore, the caption may not be compatible with this noisy POS tag sequence which leads to an overly smooth latent representation for the POS tag. Therefore, different POS tags may produce the same latent code and hence the same caption.

(2) **Novel sentences:** We measure the number of novel sentences (not seen in train), and find that our POS-based methods produce more novel sentences than all other methods. Beam search produces the least number of novel sentences.

(3) **Mutual overlap:** We also measure the mutual overlap between generated captions. This is done by taking one caption out of $k$ generated captions and evaluating the average Bleu-4 with respect to all other $k-1$ captions. Lower value indicates higher diversity. POS is the most diverse. Note, the average score is computed by picking every caption *vs*. the remaining $k-1$ captions.

(4) **n-gram diversity (div-$n$):** We measure the ratio of distinct $n$-grams per caption to the total number of words generated per image. POS outperforms other methods.

| Method | #samples | Meteor | Spice |
|---|---|---|---|
| Beam Search (with VGG-16) | 5 | .247 | .175 |
| GAN (with Resnet-152) | 5 | .236 | .166 |
| POS+Joint (with VGG-16) (Shetty et al., 2017) | 5 | **.247** | **.180** |

Table 3.5: **Comparison to GAN-based method.** To compare to GAN, we train our POS+Joint on another split of MSCOCO by Karpathy & Fei-Fei (2015). Our POS+Joint method samples more accurate best-1 captions than the GAN method. POS+Joint also obtains better SPICE score on this split compared to beam search. Our accuracy may improve with the use of Resnet-152 features. For fair comparison, we use the same 80-dimensional object vectors from faster rcnn (Ren et al., 2015) and rank the generated captions with likelihood for all methods.

| Baseline Method | POS Wins | Baseline Method Wins |
|---|---|---|
| Beam search | **57.7%** | 42.2% |
| Diverse beam search (Vijayakumar et al., 2018) | 45.3% | **54.6%** |
| AG-CVAE (Wang et al., 2017c) | **64.8%** | 35.1% |

Table 3.6: We show the user captions sampled from best-$k$ (same $k^{\text{th}}$ ranked, $k = 1$ to 5) for baseline methods and our POS. The user is allowed to pick the caption that best describes the image. Note, user is not aware of the method that generated the caption. Here, we observe that our POS method outperforms Beam search and AG-CVAE on our user study. Our user study has 123 participants with on average 23.3 caption pairs annotated by each user.

### 3.4.4 Speed

In Figure 3.1 we showed that our POS based methods have better time complexity than beam search and diverse beam search. The time complexity of our POS-based approach is

the same as sampling from a VAE or GAN, provided the max probability word is chosen at each word position (as we do). The empirical results in Table 3.2 and Table 3.3 show that POS methods are 5× faster than beam search methods.

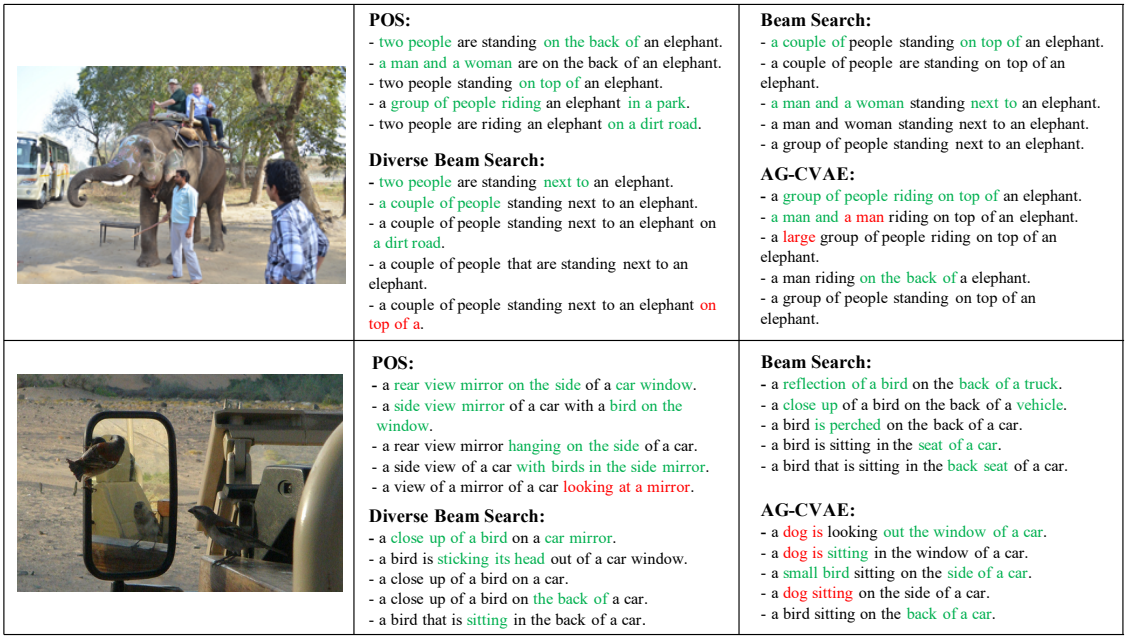



Figure 3.4: **Qualitative Comparison:** Notice POS captions contain things like rear/side view mirror, dirt road, the quantifier 'two' which is less common in other methods. The inaccuracies are highlighted in red and the novel parts in green.

### 3.4.5 User Study and Qualitative Results

Figure 3.4 compares the captions generated by different methods and in Table 3.6, we provide the results of a user study. A user is shown two captions sampled from two different methods. The user is asked to pick the more appropriate image caption. Table 3.6 summarizes our results. We observe POS outperforms AG-CVAE and Beam search. Figure 3.5 shows top 10 captions generated by POS, Beam search and Diverse beam search, for the same image and the mBleu-4 between these captions. POS performs the best.

## 3.5 RELATED WORK

In the following, we first review works that generate a single (or best-1) caption before discussing diverse image captioning methods which produce $k$ different (or a set of best-$k$) captions.

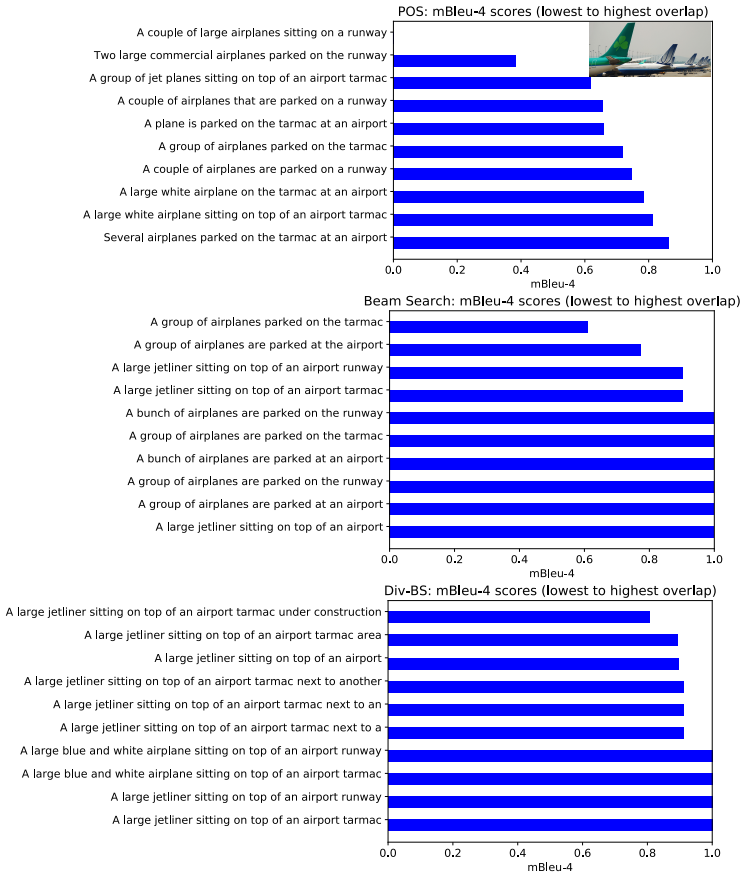



Figure 3.5: **Diversity (or Overlap) Comparison:** We compare the diversity (or overlap) of captions. The mBleu-4 score measures 4-gram overlap between one generated caption and the rest. Lower is better, *e.g.*, 0 means caption has no 4-gram overlap to other sentences. POS is better than BS and Div-BS in the plots above (lower mBleu-4 scores). Note, ground-truth 5 captions all have 0 overlap to each other for this example. On our 1000 image test set with 10 captions generated per image, POS generates 10.94% sentences with 0 overlap; in contrast Div-BS generates 1.02% and Beam Search 2.4%.

### 3.5.1 Image Captioning

Most image captioning approaches (Karpathy & Fei-Fei, 2015; Vinyals et al., 2015a; Xu et al., 2015) use a convolutional neural net pre-trained on classification Simonyan & Zisserman (2015) to represent image features. Image features are fed into a recurrent net (often based on long-short-term-memory (LSTM) units) to model the language word-by-word. These networks are trained with maximum likelihood. To obtain high performance on standard image captioning metrics, Yao et al. (2017a) use a network trained on COCO-attributes in addition to image features. Anderson et al. (2018) develop an attention-based network architecture. Aneja et al. (2018) change the language decoder from an LSTM-net

to a convoluational network and show that they obtain more diversity. Similarly, Wang & Chan (2018) also use a convolutional language decoder. Since diversity is of interest to us, we use the convolutional language decoder similar to Aneja et al. (2018); Wang & Chan (2018). We leave incorporation of techniques such as attribute vectors specific to the COCO dataset, and a sophisticated attention mechanism from Yao et al. (2017a); Anderson et al. (2018) for further performance gains to future work.

Apart from exploring different network architectures, some prior works focus on using different training losses. Reinforcement learning has been used by Luo et al. (2018); Rennie et al. (2017b); Liu & Wang (2016), to directly train for non-differentiable evaluation metrics such as BLEU, CIDEr and SPICE. In this work, we use maximum likelihood training for our methods and baselines to ensure a fair comparison. Training our POS captioning network in a reinforcement learning setup can be investigated as part of future work.

Notable advances have been made in conditioning image captioning on semantic priors of objects by using object detectors (Lu et al., 2018; Wang et al., 2018). This conditioning is only limited to the objects (or nouns) in the caption and ignores the remainder, while our POS approach achieves coordination for the entire sentence.

### 3.5.2 Diverse Image Captioning

Four main techniques have been proposed to generate multiple captions and rank them to obtain a set of best-$k$ captions.

**Beam search:** Beam search is the classical method to sample multiple solutions given sequence models for neural machine translation and image captioning. We compare to beam search on the same base captioning network as POS, but without part-of-speech conditioning. We find that though beam search is accurate, it is slow (Table 3.3) and lacks diversity (Table 4.2). Our base captioning network uses a convolutional neural net (CNN) (Aneja et al., 2018) and is equivalent to the standard LSTM based captioning network of Karpathy & Fei-Fei (2015) in terms of accuracy.

**Diverse beam search:** Vijayakumar et al. (2018) augment beam search with an additional diversity function to generate diverse outputs. They propose a hamming diversity function that penalizes expanding a beam with the same word used in an earlier beam. In our results, we compare to this diverse beam search (Div-BS). Note, beam search and diverse beam search are local search procedures which explore the output captioning space word-by-word. While, POS tag sequences act as global probes that permit to sample captions in many different parts of the captioning space.

**GAN:** More recent work on diverse image captioning focuses on using GANs. Adversarial training has been employed by Dai et al. (2017); Li et al. (2018); Shetty et al. (2017) to generate diverse captions. Dai et al. (2017); Li et al. (2018) train a conditional GAN for diverse caption generation. Dai et al. (2017) use a trainable loss which differentiates human annotations from generated captions. Ranking based techniques, which attempt to score human annotated captions higher than generated ones, are demonstrated by Li et al. (2018). Shetty et al. (2017) use adversarial training in combination with an approximate Gumbel sampler to match the generated captions to the human annotations.

Generally, GAN based methods improve on diversity, but suffer on accuracy. For example, in Tab. 1 of Shetty et al. (2017), METEOR and SPICE scores drop drastically compared to an LSTM baseline. In Table 3.5, we compare GAN (Shetty et al., 2017) and our POS-based method which is more accurate.

**VAE:** Wang et al. (2017c) propose to generate diverse captions using a conditional variational auto-encoder with an additive Gaussian latent space (AG-CVAE) instead of a GAN. The diversity obtained with their approach is due to sampling from the learned latent space. They demonstrate improvements in accuracy over the conventional LSTM baseline. Due to the computational complexity of beam search they used fewer beams for the LSTM baseline compared to the number of captions sampled from the VAE, *i.e.*, they ensured equal computational time. We compare to AG-CVAE (Wang et al., 2017c) and show that we obtain higher best-1 caption accuracy (Table 3.3) and our best-$k^{\text{th}}$ caption accuracy ($k = 1$ to 10) outperforms AG-CVAE (Figure 3.3). Note, best-$k$ scores in Table 3.3 and Figure 3.3 denote the score of the $k^{\text{th}}$ ranked caption given the same number of sampled captions (20 or 100) for all methods. For fairness, we use the same ranking procedure (*i.e.*, consensus re-ranking proposed by Devlin et al. (2015) and used by Wang et al. (2017c)) to rank the sampled captions for all methods.

## 3.6 CONCLUSION

The developed diverse image captioning approach conditions on part-of-speech. It obtains higher accuracy (best-1 and best-10) than GAN and VAE-based methods and is computationally more efficient than the classical beam search. It performs better on different diversity metrics compared to other methods.

# CHAPTER 4: SEQUENTIAL LATENT SPACES FOR MODELING THE INTENTION DURING DIVERSE IMAGE CAPTIONING

## 4.1 INTRODUCTION

Diverse yet accurate image captioning is an important goal towards augmenting present-day editing and auto-response tools with technology that maintains creative freedom while providing meaningful and inspiring suggestions. On the quest to succeed in this tightrope walk, methods need to maintain accuracy of the provided descriptions while elaborately managing the intricacies of the respective language. This balancing act to aesthetically craft short yet precise statements that hit the point is an art.

Nonetheless, any description is always and inherently targeted towards a group of readers. Consequently, the message of the description remains hard to access or even inaccessible for any other audience, because words are overloaded and the crisp picture that we intend to draw in a readers mind blurs rapidly. Yet, its efficacy relies heavily on the ability to accurately convey the main ideas.

Going forward, imagine your description to automatically adjust depending on the background knowledge of the reader. Obviously we are far from this idea being remotely feasible. Nonetheless, in recent years, remarkable progress has been made in image captioning (Johnson et al., 2016; Barnard et al., 2003; Chen & Zitnick, 2015; Donahue et al., 2015; Fang et al., 2015; Farhadi et al., 2010; Cho et al., 2015; Karpathy & Fei-Fei, 2015; Kiros et al., 2015; Kulkarni et al., 2011; Mao et al., 2015; Socher et al., 2014a; Vinyals et al., 2015a; Xu et al., 2015) and particularly controllable (Wang et al., 2017c; Deshpande et al., 2019) and diverse (Wang et al., 2017c; Deshpande et al., 2019; Dai et al., 2017; Li et al., 2018; Shetty et al., 2017; Vijayakumar et al., 2018) image captioning. Many of the proposed mechanisms for image captioning rely on long-short-term-memory (LSTM) Hochreiter & Schmidhuber (1997a) nets where words are generated one at a time. For diversity, LSTM based variational auto-encoders (Kingma et al., 2014) or generative adversarial nets (Goodfellow et al., 2014a) and their conditional counterparts (Sohn et al., 2015) are employed. For high-level control, one-hot encodings that represent observed objects or groups of objects are injected at the first step of the LSTM (Wang et al., 2017c). Recently (Deshpande et al., 2019), more low-level control has also been discussed by conditioning on abstract representations of part-of-speech tags. Again, the conditioning was achieved by changing the initial LSTM input.

Because of this single initial conditioning input, none of the aforementioned methods provide the fine-grained diversity that is desirable for adjusting individual words of a sentence.

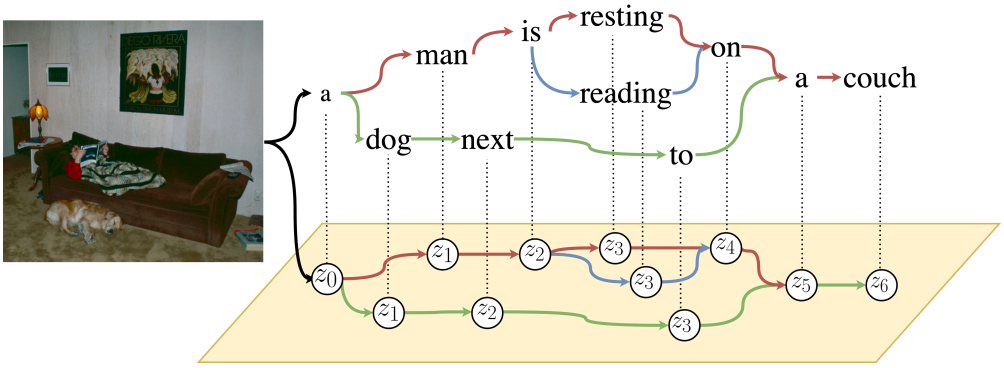



Figure 4.1: Meaningful diverse captions generated (blue arrows) for a given image by linearly interpolating from one latent vector (green arrows) to another (red arrows).

We address this shortcoming by developing **Seq-CVAE**, a method which learns a latent space for every word. Importantly, we want the latent space to be predictive of the subsequent parts of the sentence, *i.e.*, the future of the sentence. We achieve this by employing a data-dependent transition model which captures the 'intention,' *i.e.*, a representation which encodes the remaining part of the sentence. During training the intention model is tasked to fit the representations of a backward LSTM.

This proposed approach enables to distinctly modify descriptions starting from a particular position.

We demonstrate this fine-grained diversity by sampling a diverse set of captions and linearly interpolating between the chosen latent representations. As illustrated in Figure 6.1, a convex combination of latent vectors permits to gradually transition from one caption to another.

We evaluate the proposed approach quantitatively on the challenging COCO (Lin et al., 2014) dataset and significantly outperform competing methods w.r.t. diversity metrics: among 5000 sampled captions more than 4200 are novel and not part of the training set while the runner-up baselines are just short of 3500. Despite this diversity the proposed approach is on par w.r.t. accuracy metrics. These results are particularly remarkable because all competing baselines use additional information in the form of object detectors (Ren et al., 2015) during inference, while the proposed approach does not use any additional information during inference.

**Contributions:** We develop an image captioning model with a sequential latent space to capture the intention, *i.e.*, the future of the sentence (Section 4.2). We show that sampling with sequential latent spaces results in significantly more diverse captions than baselines (Table 4.2) despite not using any additional information. Perceptual metrics of our diverse captions are on par with baselines (Table 4.1). The sequential latent space permits distinct

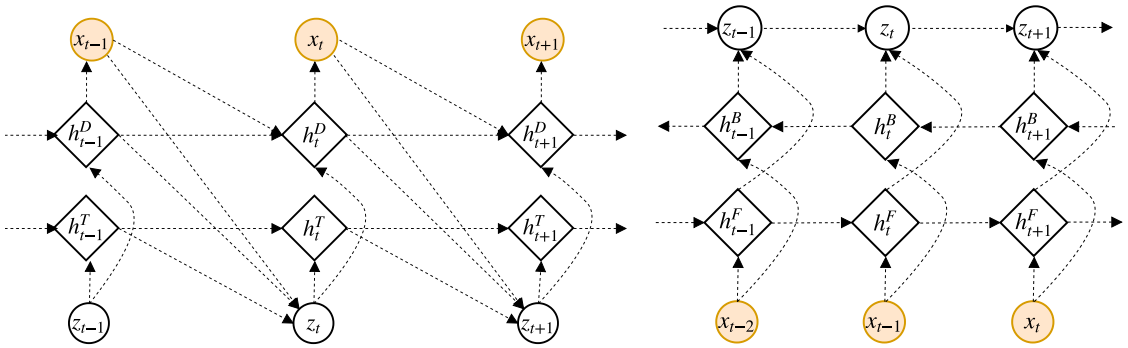



Figure 4.2: **(a)**: Computation graph for the generation network. The hidden states of the intention model LSTM and the decoder LSTM are $h_t^T$ and $h_t^D$ respectively. At a given time step $t$, the latent sample $z_t$ depends on all prior words $x_{<t}$ and all prior latent samples $z_{<t}$. The sample $z_t$ along with all prior words $x_{<t}$ predicts $x_t$. **(b)**: Computation graph for the encoder network. The hidden states of the forward LSTM and the backward LSTM are $h_t^F$ and $h_t^B$ respectively. At a given time $t$, the latent sample $z_t$ depends on the entire caption $x$ through the forward and backward models.

access to sentences starting from a specific position (Figure 6.1).

## 4.2 APPROACH

In the following we first outline the proposed approach before discussing individual components.

### 4.2.1 Overview

Given an image $I$ we are interested in generating a diverse set of captions $x^k$, $k \in \{1, \ldots, K\}$. For readability we drop the index $k$ henceforth and note that the developed method will produce many captions that are ranked subsequently. Every generated caption $x = (x_1, \ldots, x_T)$ is a tuple consisting of words $x_t \in \mathcal{X}$, $t \in \{1, \ldots, T\}$, each from a discrete vocabulary $\mathcal{X}$. Given an image $I$ we devise a probabilistic model $p_\theta(x|I)$ which depends on parameters $\theta$ and assigns a probability to every caption $x$.

To effectively sample from this probabilistic space we assume the probability distribution $p_\theta(x|I)$, jointly defined over all words $x_t$ of a caption, to factorize into a product of word-conditionals, *i.e.*,

$$p_\theta(x|I) = \prod_{t \in \{1,\ldots,T\}} p_\theta(x_t|x_{<t}, I).$$

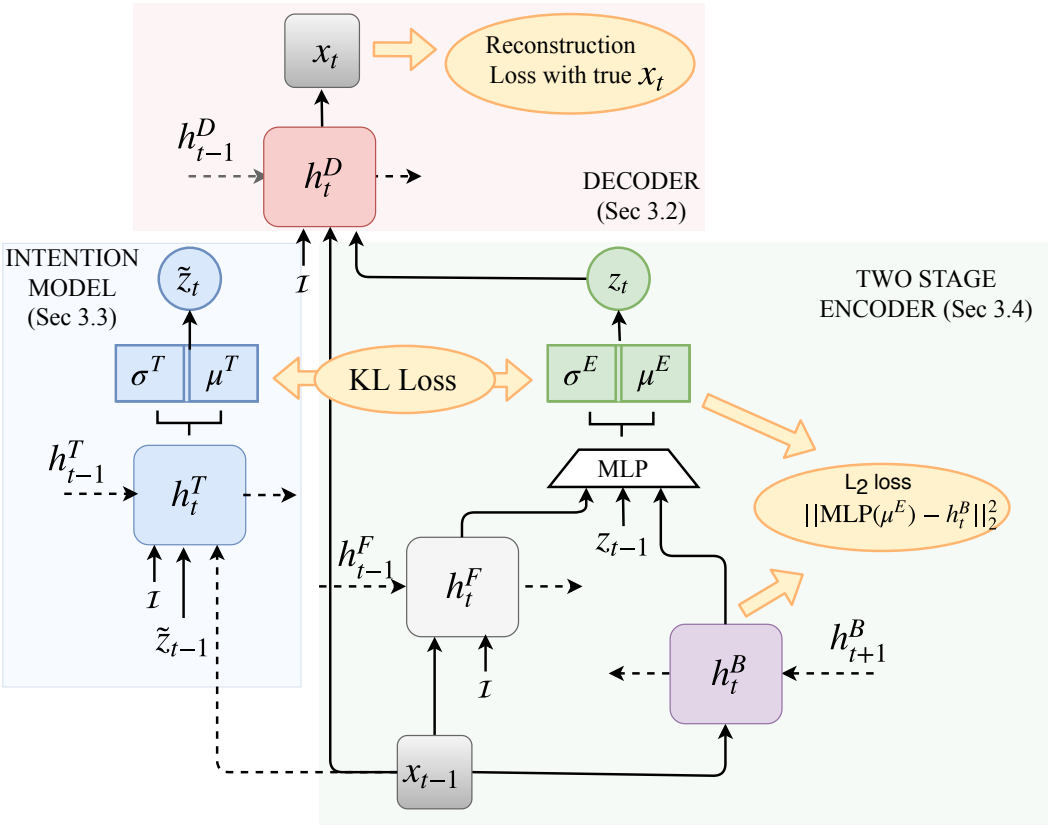



Figure 4.3: Our proposed **Seq-CVAE** training architecture, shown for a single time slice $t$. Our model includes three components during training: (1) Two-stage encoder; (2) Intention Model; and (3) Decoder. Details for each of the components are provided in Section 4.2. At test time only decoder and intention model are used.

This factorization enforces a temporal ordering, *i.e.*, the probability distribution for word $x_t$ is conditioned on all preceding words $x_{<t}$. Importantly, because the conditional's domain is the vocabulary space $\mathcal{X}$ and not a product space thereof, as it is the case for the joint distribution $p_\theta(x)$, ancestral sampling is a suitable and effective technique to generate a diverse set of captions.

In practice the conditional probability distributions $p_\theta(x_t|x_{<t}, I)$, often also referred to as the decoder distributions, are modeled via recurrent LSTM nets or masked convolutions, where we use $h^D_t$ to refer to its hidden state which summarizes $x_{<t}$, while $x_{t-1}$ directly influences the distribution. However, given the preceding words $x_{<t}$ the conditional distribution $p_\theta(x_t|x_{<t}, I)$ has to cover many words which are suitable to complete the sentence. While LSTM-nets can model complex distributions, we hypothesize that a latent variable $z_t$ which captures the current intention about how to complete the sentence can significantly reduce the complexity of the conditional.

Therefore, instead of directly modeling $p_\theta(x_t|x_{<t}, I)$ which marginalizes across all possible intentions, we are interested in modeling and sampling from the decoder distribution $p_\theta(x_t|x_{<t}, z_{\leq t}, I)$, *i.e.*, a distribution conditioned on the current and all previous intentions, which are however unobserved.

**During inference**, as illustrated in Figure 4.2 (a), we ensure effective sampling of an intention $z_t$ by modeling the transition through the tuple of all intentions $z = (z_1, \ldots, z_T)$ again via a product of conditionals $p_\theta(z_t|z_{<t}, x_{<t}, I)$, *i.e.*,

$$\text{(intention model)} \qquad \prod_{t\in\{1,\ldots,T\}} p_\theta(z_t|z_{<t}, x_{<t}, I).$$

To obtain a description for a given image, as shown in Figure 4.2 (a), we alternatingly sample from $p_\theta(z_t|z_{<t}, x_{<t}, I)$, which is sometimes referred to as the 'prior,' and from the decoder $p(x_t|x_{<t}, z_{\leq t}, I)$. Different from classical approaches we employ a parametric 'intention model' which decomposes temporally. Note that the 'intention distribution' at time $t$, *i.e.*, $p_\theta(z_t|z_{<t}, x_{<t}, I)$ is dependent on $x_{<t}$. This is technically correct due to the temporal decomposition, *i.e.*, the distribution at time $t$ can depend on all previously available data. Similar to the decoder, we use an LSTM net to capture the recurrence of the intention model and refer to its latent state via $h_t^T$.

To model the intentions $z$, during training, as illustrated in Figure 4.2 (b), we encourage the intention model to fit the approximated posterior $q_\phi(z|x, I)$ which we model using again a product of conditionals, *i.e.*,

$$\text{(approx. posterior)} \quad q_\phi(z|x, I) = \prod_{t\in\{1,\ldots,T\}} q_\phi(z_t|z_{t-1}, x, I).$$

The distribution $q_\phi(z|x, I)$ is commonly referred to as the encoder. To adequately capture the intention on how to complete the sentence, as illustrated in Figure 4.2 (b), we develop a two-stage encoder consisting of a forward stage to model language and a backward stage to summarize intention, *i.e.*, the future of the sentence. We discuss details in Section 4.2.4.

**During training** we are given a dataset $\mathcal{D} = \{(I, x)\}$ consisting of pairs $(I, x)$, each containing an image $I$ and a corresponding caption $x$. We maximize the data log-likelihood $-\sum_{(I,x)\in\mathcal{D}} \ln p_\theta(x|I)$. For readability we drop the summation over samples in the dataset subsequently. By using the aforementioned decompositions we note that the data log-

likelihood $\ln p_\theta(x|I)$ is obtained by marginalizing over the space of intentions, *i.e.*,

$$\ln p_\theta(x|I) = \ln \sum_z p_\theta(x, z|I) = \ln \sum_z \prod_t \underbrace{\overbrace{p_\theta(x_t|x_{<t}, z_{\leq t}, I)}^{\text{decoder}} \overbrace{p_\theta(z_t|z_{<t}, x_{<t}, I)}^{\text{intention}}}_{p_\theta(x_t, z_t|x_{<t}, z_{<t}, I)}.$$

Marginalization over the space of intentions makes maximization of this objective computationally expensive. It is therefore common to utilize an approximate posterior and apply Jensen's inequality which gives the lower bound

$$\ln p_\theta(x|I) \geq \mathbb{E}_{z\sim q_\phi(z|x,I)} \left[ \ln \frac{p_\theta(x, z|I)}{q_\phi(z|x, I)} \right].$$

Combined with the employed temporal decomposition, this yields the objective

$$\mathbb{E}_{z\sim q_\phi(z|x,I)} \Bigg[ \sum_t \big( \ln p_\theta(x_t|x_{<t}, z_{\leq t}, I) + \ln p_\theta(z_t|z_{<t}, x_{<t}, I) - \ln q_\phi(z_t|z_{t-1}, x, I) \big) \Bigg], \tag{4.1}$$

which we maximize w.r.t. parameters $\theta$ and $\phi$. In the following we discuss decoder, prior (intention model) and encoder in more detail. Notice all their parameters are subsumed in the vectors $\theta$ and $\phi$ and jointly trained end-to-end. A single timestep of all three recurrent models is illustrated in Figure 4.3.

### 4.2.2 Decoder

As illustrated in the top part of Figure 4.3, the decoder is a classical LSTM net. At time $t$ the decoder yields a multinomial probability distribution $p_\theta(x_t|x_{<t}, z_{\leq t}, I)$ defined over words $x_t \in \mathcal{X}$. While representations of $z_t$ and $x_{t-1}$ are concatenated before being provided as input to the LSTM net, we encode dependence on $x_{<t-1}$ and $z_{<t}$ via its hidden representation $h^D_{t-1}$. Dependence on the image is encoded into the LSTM net via an image embedding obtained from the fc7 layer of a VGG16 network (Simonyan & Zisserman, 2015), pre-trained on the ImageNet dataset. The image embedding is fed as input at every time step of the LSTM, concatenated with the input word embedding and the sampled vector from the latent space. The latent vector dimension can be chosen as 512, 256 or 128. For all our experiments, we found the size of the latent vector to be 512 to give the best results. Table 4.4.

### 4.2.3 Intention Model

Similar to the decoder we model the intention transition model $p_\theta(z_t|z_{<t}, x_{<t}, I)$ as an LSTM net. However, different from the decoder, given $z_{<t}$ and $x_{<t}$ we model $p_\theta(z_t|z_{<t}, x_{<t}, I)$ as a Gaussian distribution with time-dependent mean $\mu_t^T(z_{<t}, x_{<t}, I)$ and standard deviation $\sigma_t^T(z_{<t}, x_{<t}, I)$ obtained from an LSTM net. The LSTM net input $z_{t-1}^T$ and $x_{t-1}^T$ directly influence $\mu_t^T$ and $\sigma_t^T$. Dependence on $z_{<t-1}$ and $x_{<t-1}$ is encoded via the hidden representation $h_{t-1}^T$. Dependence on the image is encoded into the LSTM net via an image embedding obtained from the fc7 layer of a VGG16 network (Simonyan & Zisserman, 2015), pre-trained on the ImageNet dataset. The 512 dimensional image embedding is fed as input at every time step of the intention model LSTM, concatenated with the output from the previous time step and the word embedding of the previous word. The image embedding, the word embedding and the latent vector are all 512 dimensional.

During inference, at time $t$ we use a sample $z_t$ from the modeled Gaussian with mean $\mu_t^T(z_{<t}, x_{<t}, I)$ and standard deviation $\sigma_t^T(z_{<t}, x_{<t}, I)$ as input for the decoder. However, during training, as illustrated in Figure 4.3, we use a sample from the encoder. This discrepancy is justified by the fact that part of the training objective given in Eq. (4.1) maximizes the negative KL-divergence

$$\sum_t -\mathbb{E}_{z_t \sim q_\phi(z_t|z_{t-1}, x, I)} \left[ \ln \frac{q_\phi(z_t|z_{t-1}, x, I)}{p_\theta(z_t|z_{<t}, x_{<t}, I)} \right]$$

between the intention model and the encoder at each time-step. This is highlighted in Figure 4.3. Therefore, upon training we want those distributions to be adequately close, which ensures that the samples used during testing are suitable.

More importantly however, note that the encoder $q_\phi(z_t|z_{t-1}, x, I)$ depends on the entire sentence $x$ while the intention model $p_\theta(z_t|z_{<t}, x_{<t}, I)$ only depends on the past observations $x_{<t}$. Consequently, if we construct an adequate encoder and if $p_\theta(z_t|z_{<t}, x_{<t}, I)$ is accurate, we are able to capture the intention about how to complete the sentence using samples from the intention model. We discuss an encoder structure that yielded encouraging results next.

### 4.2.4 Encoder to Expose Intention

To adequately encode the intention, *i.e.*, the future of a sentence, we need to construct a model which contains at time $t$ information about the entire sentence rather than only its past. To achieve this we develop a two-stage encoder $q_\phi(z_t|z_{t-1}, x, I)$ which models at time $t$ a Gaussian distribution with mean $\mu_t^E(z_{t-1}, x, I)$ and standard deviation $\sigma_t^E(z_{t-1}, x, I)$

modulated by multiplying with the exponentiated function value $F_\phi(\mu_t^E, x, I) \in \mathbb{R}$, *i.e.*,

$$q_\phi(z_t|z_{t-1}, x, I) \propto \mathcal{N}(z_t|\mu_t^E, \sigma_t^E) \cdot \exp F_\phi(\mu_t^E, x, I).$$

Note that multiplication with the exponentiated function value $F_\phi(\mu_t^E, x, I)$ doesn't change the fact that $q_\phi$ is a valid distribution over the latent space. Importantly however, multiplication permits to add the term $-F_\phi(\mu_t^E, x, I)$ to the objective given in Eq. (4.1). As detailed below, we will use $F_\phi$ to encourage $\mu_t^E$ to better capture the future of a sentence.

The first stage of the encoder captures the past via a classical forward LSTM net with hidden states referred to as $h_t^F$. The second stage of the encoder captures the future via a backward LSTM net with hidden states referred to as $h_t^B$. We subsequently combine both via a multi-layer perceptron (MLP) net which yields mean $\mu_t^E(z_{t-1}, x, I)$ and standard deviation $\sigma_t^E(z_{t-1}, x, I)$ of the Gaussian distribution.

To ensure that the mean $\mu_t^E(z_{t-1}, x, I)$ more closely resembles the information obtained from the backward pass we choose

$$F(\mu_t^E, x, I) = \lambda \|g(\mu_t^E(z_{t-1}, x, I)) - h_t^B\|_2^2, \tag{4.2}$$

where $g$ is another MLP which maps $\mu_t^E$ to fit $h_t^B$. The latter is 512-dimensional in our case. $\lambda$ is a hyper-parameter set to $5e^{-4}$. For the backward LSTM we use the pre-trained ELMo (Peters et al., 2018) model, with a hidden dimension of 512. ELMo is a deep bidirectional language model trained on 1 Billion Word Language Model Benchmark (Chelba et al., 2013). We only use the backward part of the model. Word representations taken from this backward pass at any time $t$ is a good encoding of the all the $x_{>t}$ . ELMo is not fine-tuned through training.

### 4.3 RESULTS

In the following, we first describe the dataset along with the competitive baselines for diverse captioning and the evaluation metrics used. We then present our results.

**Dataset:** We use the challenging **MS COCO** dataset (Lin et al., 2014) for our experiments. Following the approach by Deshpande et al. (2019); Wang et al. (2017d), we perform our analysis on

the split of M-RNN (Mao et al., 2015) which has 118,287 train, 4,000 val and 1,000 test images and defer additional results to the supplementary material.

**Methods:** We refer to our proposed approach as **Seq-CVAE**. We also provide results

| | Method | Best-1 Oracle Accuracy on M-RNN Test Split | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|
| | | B4 | B3 | B2 | B1 | C | R | M | S |
| Beam size #samples: 20 | Beam search | **0.489** | **0.626** | **0.752** | **0.875** | **1.595** | **0.698** | **0.402** | **0.284** |
| | Div-BS (Vijayakumar et al., 2018) | 0.383 | 0.538 | 0.687 | 0.837 | 1.405 | 0.653 | 0.357 | 0.269 |
| | AG-CVAE (Wang et al., 2017d) | 0.471 | 0.573 | 0.698 | 0.834 | 1.259 | 0.638 | 0.309 | 0.244 |
| | POS (Deshpande et al., 2019) | 0.449 | 0.593 | 0.737 | 0.874 | 1.468 | 0.678 | 0.365 | 0.277 |
| | Seq-CVAE | 0.445 | 0.591 | 0.727 | 0.870 | 1.448 | 0.671 | 0.356 | 0.279 |
| Beam size #samples: 100 | Beam Search | **0.641** | **0.742** | **0.835** | **0.931** | **1.904** | **0.772** | **0.482** | **0.332** |
| | Div-BS (Vijayakumar et al., 2018) | 0.402 | 0.555 | 0.698 | 0.846 | 1.448 | 0.666 | 0.372 | 0.290 |
| | AG-CVAE (Wang et al., 2017d) | 0.557 | 0.654 | 0.767 | 0.883 | 1.517 | 0.690 | 0.345 | 0.277 |
| | POS (Deshpande et al., 2019) | 0.578 | 0.689 | 0.802 | 0.921 | 1.710 | 0.739 | 0.423 | 0.322 |
| | Seq-CVAE | 0.575 | 0.691 | 0.803 | 0.922 | 1.695 | 0.733 | 0.410 | 0.320 |

Table 4.1: **Best-1-Oracle Accuracy.** Our **Seq-CVAE** method obtains high scores on standard captioning metrics. We obtain comparable accuracy when compared to both the very recently proposed POS approach (Deshpande et al., 2019) which uses a part-of-speech prior and also to the AG-CVAE method (Wang et al., 2017d). Both these methods use additional information in the form of object vectors from a Faster-RCNN (Ren et al., 2015) during inference. In contrast, we do not use any additional information during inference. To calculate the best-1-accuracy, we report the caption with highest CIDEr score from all the sampled captions (# samples = 20 or 100). Beam search, although obtaining the highest CIDEr score, is known to be extremely slow and significantly less diverse.

of our proposed approach without the ELMo based backward LSTM and without the data-dependent intention model. We compare the results to the diverse captioning approach of Deshpande et al. (2019) which uses part-of-speech as a prior and refer to the method via **POS**. We also compare to the additive Gaussian conditional VAE-based based diverse captioning method of Wang et al. (2017c), denoted by **AG-CVAE** which uses object detections as additional information. Moreover, we compare to beam search denoted as **Beam search** and diverse beam search (Vijayakumar et al., 2018) referred to as **Div-BS** applied to standard captioning methods based on convolutions (Aneja et al., 2018) and LSTM nets (Karpathy & Fei-Fei, 2015).

**Evaluation criteria:** We compare all aforementioned methods via the following accuracy and diversity metrics:

- **Accuracy.** In Section 4.3.1, we report the Top-1-Accuracy evaluated on the standard image captioning metrics (Bleu-1 to Bleu-4, CIDEr, ROUGE, METEOR and SPICE, each abbreviated with its initial).
- **Diversity.** Our diversity evaluation is presented in Section 4.3.2

### 4.3.1 Top-1-Accuracy

We use the CIDEr as an oracle to pick the top-1 caption from a set of generated diverse captions for a given image.

Following the approach of Deshpande et al. (2019) and Wang et al. (2017c), the top-1 caption is chosen as that caption with the maximum score calculated with the ground-truth test captions as references. The oracle metric provides the maximally possible top-1 accuracy that a given model can achieve. In Table 4.1 we show that our **Seq-CVAE** performs on par with the best baselines on the MRNN split (Mao et al., 2015). Specifically, in Table 4.1 we show for 20 and 100 samples, that the proposed approach obtains a CIDEr of 1.448 and 1.695 respectively, on par with POS. Importantly, we emphasize that the proposed approach doesn't use any additional information in the form of part-of speech tags or object vectors from a Faster-RCNN (Ren et al., 2015) during inference. Note that all the other scores, are comparable to the POS (Deshpande et al., 2019) approach while improving upon the AG-CVAE (Wang et al., 2017d) method, which is the only other VAE based method which exhibits stochasticity when producing diverse captions. Note that although beam search obtains the best scores, it is known to be slow and less diverse as shown in Section 4.3.2.

### 4.3.2 Diversity Evaluation

To ensure comparability to the baselines, our diversity numbers are calculated on the M-RNN split (Mao et al., 2015). In Table 4.2 we compare our performance on diverse generation with the baselines, using the following metrics:

**(1) Uniqueness.** The number of distinct captions generated by sampling from the latent space. We show that the proposed method produces 18.48/20 (92.4%) and 80.9/100 (80.9%) unique sentences. Note that beam search and Div-BS are deterministic and are guaranteed to generate 100% unique captions. Similarly, POS is completely deterministic and ensures a large number of unique captions via a strong connection between generated words and a hard-coded 'latent space' which depends on part-of-speech tags and is learned in a fully supervised manner. In contrast, AG-CVAE, just like the proposed approach, has a flexible latent space. Compared to AG-CVAE, the proposed approach generates significantly more distinct captions.

**(2) Novel Sentences.** Novel sentences are those sentences which were never observed in the training data. We see that **Seq-CVAE** produces significantly more novel sentences than any other baseline. This is remarkable and illustrates the ability to emit novel words that form reasonable sentences, particularly when considering that accuracy metrics are on par

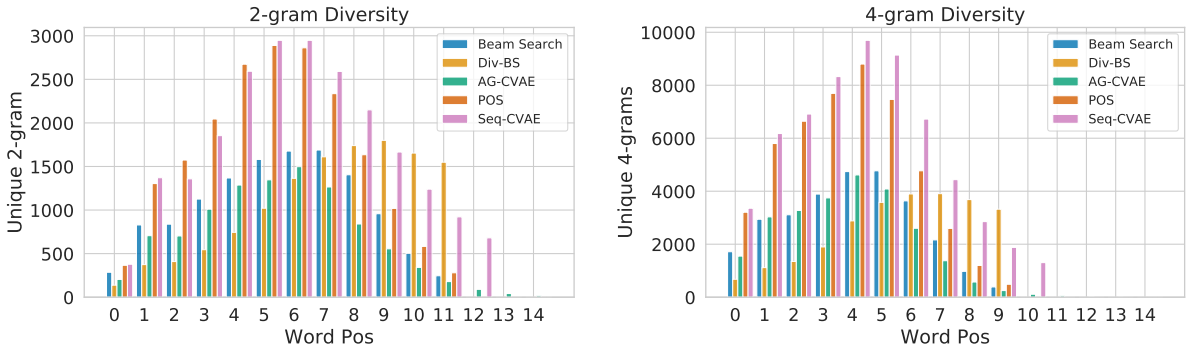
| | Method | Distinct Captions | # Novel Sentences | mBleu-4 | $n$-gram Diversity | |
|---|---|---|---|---|---|---|
| | | | | | Div-1 | Div-2 |
| Beam size #samples: 20 | Beam search | **100%** | 2317 | 0.77 | 0.21 | 0.29 |
| | Div-BS (Vijayakumar et al., 2018) | **100%** | 3106 | 0.81 | 0.20 | 0.26 |
| | AG-CVAE (Wang et al., 2017d) | 69.8% | 3189 | 0.66 | 0.24 | 0.34 |
| | POS (Deshpande et al., 2019) | 96.3% | 3394 | 0.64 | 0.24 | 0.35 |
| | Seq-CVAE | 94.0% | **4266** | **0.52** | **0.25** | **0.54** |
| Beam size #samples: 100 | Beam search | **100%** | 2299 | 0.78 | 0.21 | 0.28 |
| | Div-BS (Vijayakumar et al., 2018) | **100%** | 3421 | 0.82 | 0.20 | 0.25 |
| | AG-CVAE (Wang et al., 2017d) | 47.4% | 3069 | 0.70 | 0.23 | 0.32 |
| | POS (Deshpande et al., 2019) | 91.5% | 3446 | 0.67 | 0.23 | 0.33 |
| | Seq-CVAE | 84.2% | **4215** | **0.64** | **0.33** | **0.48** |
| 5 | Human | 99.8% | - | 0.51 | 0.34 | 0.48 |

Table 4.2: **Diversity Statistics.** We report the number of novel sentences (sentences never seen during training) for each method. Beam search and diverse beam search (Div-BS) produce the least number of novel sentences. POS (Deshpande et al., 2019) uses additional information in the form of part-of-speech tokens and object detections from Faster-RCNN (Ren et al., 2015). AG-CVAE (Wang et al., 2017d) also uses additional information in the form of object vectors. Our **Seq-CVAE** with ELMo doesn't use any additional information during inference and produces 4278/5000 novel sentences. Our method also yields to significant improvements on 2-gram diversity, producing $\approx 20\%$ more unique 2-grams for 20 samples and a $\approx 15\%$ improvement for 100 samples when compared to the runner-up, *i.e.*, POS (Deshpande et al., 2019). The model also provides the lowest m-Bleu-4, which shows that for each image the diverse captions are most different from each other. Beam search has the highest m-Bleu-4, which shows that all the distinct captions don't differ from each other at many word locations.

with the best performing baselines. In Table 4.2, we show that our approach produces, ¿4000 novel captions among the 5000 captions chosen. We choose the top-5 generated captions per image, ranked by CIDEr, using consensus re-ranking following the approach in Devlin et al. (2015); Wang et al. (2017d).

**(3) Mutual Overlap – (mBleu-4).** m-Bleu-4 measures the difference between predicted diverse captions. Specifically, for a given image, we calculate the Bleu-4 metric for every one of the $K$ diverse captions w.r.t. the remaining $K-1$ and average across all test images. A lower value of m-Bleu indicates more diversity. Again we observe that the proposed approach significantly improves upon all baselines. Again, we use top-5 generated captions, ranked by CIDEr, using consensus re-ranking (Devlin et al., 2015; Wang et al., 2017d).

**(4) n-gram diversity – (Div-$n$):** For Div-$n$ scores, we measure the ratio of distinct $n$-grams to the total number of words generated per set of diverse captions. Higher is better.




(a) (b)

Figure 4.4: (a & b): n-gram diversity across word positions. **Seq-CVAE** improves significantly upon many baselines.

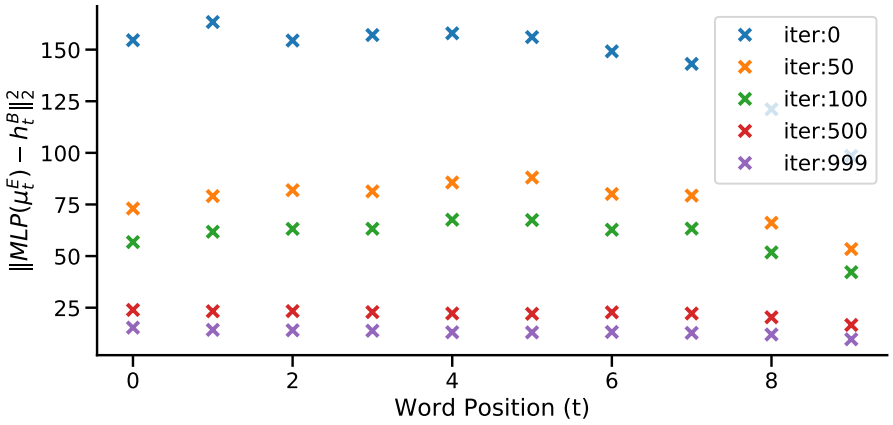



Figure 4.5: $L_2$ distance between the ELMo hidden state and the representation inferred by passing the encoder mean $\mu_t^E$ through an MLP. The latter matches $h_t^B$ better as the training progresses. This indicates that the latent space learned by the encoder at a given time $t$ is trained to better regress to word representations which summarize future words.

Again we observe that our approach significantly improves upon the baselines, particularly when considering 2-grams. For instance, we improve from 0.35 to 0.54 when considering 20 samples and from 0.33 to 0.48 when considering 100 samples. This is encouraging because it again illustrates the ability of our approach to produce fitting yet diverse descriptions without using any additional information.

**(5) Unique n-grams.** We measure the unique 2-grams and the unique 4-grams produced by our model in Figure 4.4 (a, b). We observe that our model produces the largest number of 4-grams for all word positions until position 8. We produce a comparable number of unique 2-grams as POS (Deshpande et al., 2019). To compute the numbers we use 20 samples from

| Method | ELMo | Distinct Captions | # Novel Sentences | mBleu-4 | $n$-gram Diversity | | CIDEr |
|---|---|---|---|---|---|---|---|
| | | | | | Div-1 | Div-2 | |
| POS (Deshpande et al., 2019) | - | **96.3%** | 3394 | .64 | .24 | .35 | **1.468** |
| Z-forcing (Goyal et al., 2017) | ✓ | 47.7% | **4361** | .79 | **.25** | .37 | 1.14 |
| CVAE | × | 12.1% | 1991 | .52 | .16 | .29 | .959 |
| CVAE | ✓ | 11.9% | 1923 | .51 | .25 | .29 | .952 |
| Seq-CVAE+$\mathcal{N}$ | × | 19.7% | 2888 | .63 | .24 | .35 | 1.057 |
| Seq-CVAE+$\mathcal{N}$ | ✓ | 52.8% | 4162 | .69 | .25 | .43 | 1.244 |
| Seq-CVAE(BRNN) | × | 91.8% | 4267 | .65 | **.25** | **.52** | 1.348 |
| Seq-CVAE | ✓ | 94.0% | 4266 | **.52** | **.25** | **.54** | 1.448 |
| Human | - | 99.8% | - | .510 | .34 | .48 | - |

Table 4.3: Diversity and best-1 oracle accuracy on MRNN test split for different models calculated using top-5 captions and consensus reranking.

the latent space for each of the 1000 test images. This higher number of unique 4-grams, indicates that a model is not just producing unique words, but also unique combinations of words.

### 4.3.3 Ablation Study

**(1) Is ELMo the cause of high performance?** To analyse if the improvements are only coming from a stong language model like ELMo, we replaced the ELMo with a backward RNN trained on MS-COCO training data. The performance in both diversity metrics and CIDEr, are comparable to our ELMo based model, indicating the high performance gains are not just from using a strong pretrained language model (Tab. 4.3 row 7, Seq-CVAE(BRNN)).

**(2) Using a single latent variable:** Accuracy and diversity drop when using a single $z$ (both with and without ELMo) to encode the entire sentence (Tab. 4.3 rows 3, 4; CVAE), due to posterior collapse. Also, the latent space differs per word (Figure 4.8, Figure 4.7). A single $z$ doesn't efficiently encode this.

**(3) Using a single LSTM for Encoder, Decoder and Transition Network:** Different distributions are rich are best served by their own individual representation. Following the approach by Goyal et al. (2017) using the same LSTM for all networks leads to inferior results.(Tab. 4.3 row 2, Z-forcing)

**(4) Using a constant Gaussian Distribution per word:** Replacing LSTM based learnable the intention model with a constant gaussian leads to a decline in the performance (both with and without ELMo) indicating the importance to distil intent via the backward LSTM into a sequential latent space. (Tab. 4.3 rows 5, 6; Seq-CVAE+$\mathcal{N}$)

| Image | Seq-CVAE | POS | AG-VAE | Div-BS | Beam Search |
|---|---|---|---|---|---|
| | • a cat is sitting on a suitcase on a bed<br>• cat sitting on a piece of luggage<br>• a small cat sitting on the back of a suitcase | • a cat is sitting in a suitcase on a bed<br>• a cat sitting on top of a suitcase on a bed<br>• a cat that is laying down on a suitcase | • a small gray and white cat sitting in a suitcase<br>• a white and white cat with a suitcase<br>• a cat sitting on a piece of luggage. | • a black and white cat is sitting in a suitcase<br>• a black and white cat sitting on top of a piece of luggage<br>• a cat is sitting in a suitcase | • a black and white cat is laying in a suitcase<br>• a close up of a cat laying on a luggage bag<br>• a cat that is laying down on a suitcase |
| | • the birds are swimming in the water and one is on the top<br>• two birds are standing in the water and drinking<br>• a group of birds on some water near water | • two white birds are standing in the water<br>• two large white birds standing in the water<br>• two birds are standing in the water together | • a large white and black bird in<br>• a body of water<br>• a white and white bird standing on top of a body of water<br>• a red and white photo of some birds in a pond. | • a couple of birds standing on top of a river<br>• a couple of birds standing on top of a pond<br>• a couple of birds that are standing next to each other | • couple of birds standing in the water<br>• a couple of birds that are standing in the water<br>• a couple of birds standing on a body of water |

Figure 4.6: Qualitative results illustrating captions obtained from different image captioning methods.

**(5) Conditioning over different $z$ and $x$:** The results are summarized in Table 4.4 and averaged over 10 runs. We show results without using the ELMo based backward LSTM in the encoder (see column titled ELMo), without using a data-dependent intention model ($z_t|z_{<t}$, *i.e.*, the intention model isn't conditioned on $x_{<t}$), and without using any intention (*i.e.*, the intention model is a zero mean unit variance Gaussian $\mathcal{N}$ for all word positions $t$).

Note, based on the CIDEr metric, data dependent intention doesn't contribute much when sampling 20 captions. However, data dependent intention has a slight edge when sampling 100 captions, irrespective of the chosen latent dimension. Note that the standard deviations shown in Table 4.4 are fairly small.

| Method | ELMo | Intention | Latent | C@20 | C@100 |
|---|---|---|---|---|---|
| Seq-CVAE | × | $\mathcal{N}$ | 512 | 1.015 ± 0.002 | 1.082 ± 0.001 |
| Seq-CVAE | × | $z_t\|z_{<t}$ | | 1.016 ± 0.002 | 1.089 ± 0.002 |
| Seq-CVAE | ✓ | $z_t\|z_{<t}$ | 512 | 1.332 ± 0.002 | 1.568 ± 0.004 |
| Seq-CVAE | ✓ | $z_t\|z_{<t}, x_{<t}$ | | 1.332 ± 0.002 | 1.573 ± 0.002 |

Table 4.4: **Ablation Analysis.** We observe that using the ELMo based representation improves the oracle CIDEr @100 by ∼0.5. Using ELMo along with a data dependent intention model gives the best performance with CIDEr ∼ 1.573. The low value of standard deviation calculated over 10 runs for all the models is indicative that the learned latent space is robustly structured. Using a constant Gaussian intention model ($\mathcal{N}$) performs on par with using a parametric LSTM based intention model without ELMo, clearly showing the efficacy of the proposed approach.

### 4.3.4 Latent Space Analysis

To further understand the intricacies of the learned latent space we analyze its behavior in the following.

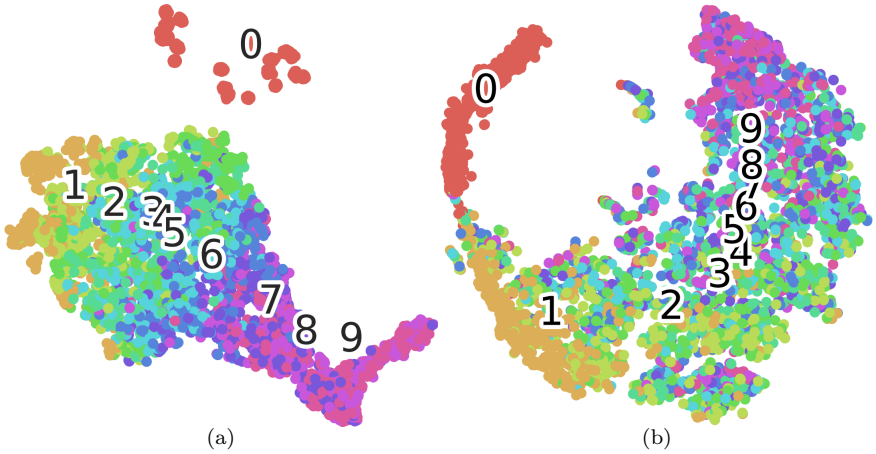



Figure 4.7: t-SNE plots of the means $\mu_t^T$ obtained from the intention model, learned **with ELMo (a)** and **without ELMo (b)**. Notice that using the ELMo representation, the model learns to better disentangle the means per word.

In Figure 4.5 we illustrate for different training iterations (see legend) and word positions $t$ the averaged $F(\mu_t^E, x, I)$ given in Eq. (4.2), *i.e.*, the L2 distance between the ELMo representation $h_t^B$ obtained via the backward LSTM and the ELMo representation inferred by passing the encoder mean $\mu_t^E$ through the MLP $g$. Intuitively we observe models at later iterations to better match the ELMo representation $h_t^B$.

To further investigate whether the mean $\mu_t^T$ of the intention model used during inference captures meaningful transitions, we illustrate in Figure 4.7 (a, b) t-SNE (van der Maaten & Hinton, 2008) plots of means obtained from different images and colored by word position $t$. We can clearly observe that the word positions of $\mu_t^T$ are much better grouped when using ELMo representation (Figure 4.7 (a)) whereas they are more cluttered when training **Seq-CVAE** without ELMo representations. We verified this analysis across multiple runs.

In Figure 4.8 we illustrate a t-SNE plot of $\mu_t^T$ based on words emitted by the decoder. We clearly observe clusters of words like 'woman,' 'man,' 'dog,' 'horse,' 'group,' 'bathroom,' 'toilet,' *etc*. This grouping is encouraging as it illustrates how we can control individual emitted words by transitioning from one representation to another. Results for this transition are illustrated in Figure 6.1.

### 4.3.5 Qualitative Results

We show a transition between two sampled captions in Figure 4.9. We linearly interpolate the latent vectors at all word positions between two sampled descriptions.

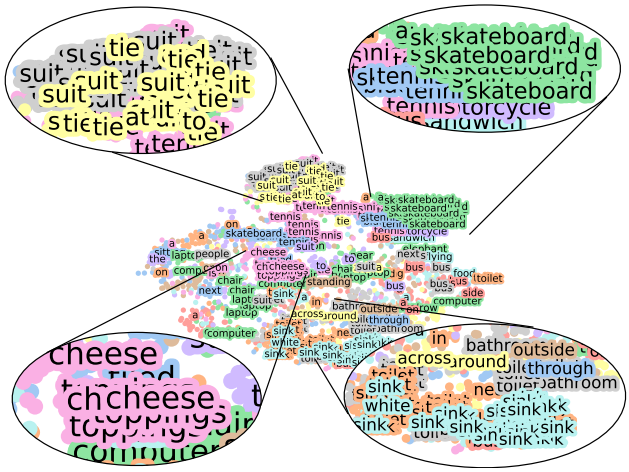


Figure 4.8: t-SNE plot of the means $\mu_t^T$ learned by the intention model, mapped to the words produced by the decoder. Notice that similar words like 'suit,' 'cheese,' *etc.* are grouped in tight clusters.

| alpha | | | |
|---|---|---|---|
| 0.0 | a horse is running down a dirt path | a boat in the water with a boat in the background | a man and woman are playing video games together |
| 0.2 | a horse is being led by a man on a horse | a boat that is floating on a body of water | people are sitting on a couch with their feet up to the wii |
| 0.4 | a man riding a horse through a field | a boat is traveling down a river | a group of people are sitting on a couch with a wii remote in |
| 0.6 | a woman walking across a dirt field with a horse | a boat is sailing in the water with a man in the background | people are sitting on a couch while playing a video game |
| 0.8 | a man riding a horse on a dirt field | a boat is floating in the water on a sunny day | a group of people are playing a video game |
| 1.0 | a woman riding a horse on a dirt field | a boat that is in the water | a group of people playing a video game together |

Figure 4.9: Diversity of sentences controlled by linear interpolation between two samples. We observe meaningful sentences across all interpolated positions.

## 4.4 RELATED WORK

Image captioning and paragraph generation (Johnson et al., 2016; Barnard et al., 2003; Chen & Zitnick, 2015; Donahue et al., 2015; Fang et al., 2015; Farhadi et al., 2010; Cho et al., 2015; Karpathy & Fei-Fei, 2015; Kiros et al., 2015; Kulkarni et al., 2011; Mao et al., 2015; Socher et al., 2014a; Vinyals et al., 2015a; Xu et al., 2015) have attracted a significant amount of work. Early classical approaches are based on sentence retrieval (Barnard et al., 2003): the best fitting sentence from a set of possible descriptions is recovered by matching sentence representations with image representations. Those representations are learned from a set of available captions. However, firstly, this matching procedure is computationally expensive and, secondly, it seems prohibitive to construct a database of captions sufficiently large to describe a reasonably comprehensive set of images accurately.

**Image Captioning:** Therefore, more recently, recurrent neural networks (RNNs) and vari-

ants like long-short-term-memory (LSTM) (Hochreiter & Schmidhuber, 1997a) networks decompose the caption space into a product space of individual words. More specifically, image representations are first extracted via a convolutional deep network which are subsequently used to prime the LSTM based recurrent network. The latter is trained via maximum likelihood to predict the next word given current and past sentence. Extensions involve object detectors (Yao et al., 2017a), attention-based deep networks (Anderson et al., 2018), and convolutional approaches (Aneja et al., 2018). Beyond maximum likelihood, reinforcement learning based techniques have also been discussed to produce a single caption, directly optimizing perceptual metrics (Liu et al., 2017; Rennie et al., 2017a). All these methods have demonstrated compelling results and have consequently been adopted widely. However, multiple captions can accurately describe an image. Consequently, diversity based methods have very recently been discussed.

**Diversity in Image Captioning:** To achieve diversity, four techniques have been investigated. Among the first was beam search, a classical approach to sample multiple captions which are assigned a high probability by the underlying model. While multiple captions are readily available, results usually only differ slightly because single word changes affect the sentence probability minimally.

To address this concern, diverse beam search (Vijayakumar et al., 2018) augments beam search to advocate for more drastic changes by encouraging to recover different modes of a probability distribution rather than high-probability configurations.

To avoid sampling from a distribution defined over a high-dimensional output space, generative adversarial networks (GANs) have been proposed (Dai et al., 2017; Li et al., 2018; Shetty et al., 2017). While GAN based methods improve on diversity, they tend to suffer on perceptual metrics.

Variational auto-encoders (VAEs) are a fourth direction that has been explored (Wang et al., 2017d). The intuition is identical to the one of GANs, *i.e.*, avoid sampling from a distribution defined over a high-dimensional output space. However, in contrast to GAN based methods, VAE based image captioning techniques tend to produce high-quality captions when evaluated on perceptual metrics.

Similar to the aforementioned approaches we develop an approach based on VAEs. However, different from all the aforementioned techniques we also aim at incorporating more fine-grained diversity.

**Controllability in Image Captioning:** Beyond diversity, controllability of captions has become an important topic very recently. In particular Wang et al. (2017c) use a variational auto-encoder conditioned on object detections to control diversity. While intuitive, control remains indirect as the sentence generating decoder is only influenced at its first timestep.

Influencing subsequent generation of words did not significantly change the result. Even more recently POSCap (Deshpande et al., 2019) was developed. While also only priming the decoder at its first step, use of clustered part-of-speech tags was proposed and shown to improve diversity. However, due to use of encoded and clustered part-of-speech tags, controllability was limited.

In contrast to the aforementioned techniques, we develop a VAE based technique which learns a latent space for every word position. While enabling diversity, this also permits direct control over words emitted at a particular position as illustrated in Figure 6.1.

**Sequential VAE:** Our proposed approach is related to a sequence of papers on sequential recurrent nets. (Fraccaro et al., 2016) develop SRNN, (Chung et al., 2015) devise VRNN, and (Bayer & Osendorfer, 2014) discuss STORN. Although VRNN (Bayer & Osendorfer, 2014), SRNN (Chung et al., 2015), Z-forcing (Goyal et al., 2017) have similar intuition, i.e., maximizing a lower bound of the data likelihood, models differ in assumptions for the prior, approximating posterior, and the decoder networks:

(1) *VRNN* uses a filtering posterior, i.e. the latent distribution at each time step depends on all the previous latent vectors and the previous input data. Instead we use a smoothing posterior, where the lantent distribution at a given time depends on the latent vector from just the previous time step and all the input data from all time steps. This leads to better models, since all context is provided.

(2) *SRNN* uses a smoothing posterior via a backward RNN as we do. However, decoder and prior differ: (a) unlike us, SRNN doesn't use latent variables in the autoregressive decoder, hence intention isn't available; (b) SRNN uses a Markovian prior, while we include the entire history of latent variables for the prior at time $t$.

(3) *Z-forcing* assumptions are similar, but differ architecturally: the prior, decoder and the approximating posterior share the same forward LSTM. This is undesirable since different distributions are best served by their own individual representation.

More crucially, these methods assess test set log-likelihood for sequential modeling, or perplexity on the IMDB dataset.

In contrast, along with accuracy, we also care about **diversity** for image captioning. Hence, we are the first to extensively study these models on various measures of diversity. (Table 4.2, Figure 4.4, Figure 4.9)

## 4.5 CONCLUSION

We propose **Seq-CVAE** which learns a word-wise latent space that captures the future of the sentence, *i.e.*, the 'intention' about how to complete the image description. This differs

from existing techniques which generally learn a single latent space to initialize sentence generation or to identically bias word generation throughout the process. We demonstrate the proposed approach on the standard dataset and illustrate results on par w.r.t. baseline accuracies while significantly improving a large variety of diversity metrics.

# CHAPTER 5: DIVERSITY UNDER THE RADAR

## 5.1 INTRODUCTION

Image captioning is an important and challenging sequential generative modeling task. Although the task description is straightforward – given an image, produce a suitable description – designing adequate models to address this task remains a difficult proposition. This is because a desirable caption generator must simultaneously satisfy at least two competing objectives: *accuracy* and *diversity*. The accuracy requirement ensures that the produced caption is syntactically correct, as dictated by the language rules, and also correlated with the image input – the caption should discuss objects depicted in the image. The diversity requirement captures the fact that image captioning is inherently an ambiguous task: the same image is accurately described via many different sentences.

Many of the proposed mechanisms for image captioning focus on the accuracy aspect and use recurrent neural nets (RNNs) and long-short-term-memory (LSTM) units. These models are trained to maximize the likelihood of human-provided ground-truth captions (Karpathy & Fei-Fei, 2015; Vinyals et al., 2015b; Cho et al., 2014; Aneja et al., 2018). Model accuracy is assessed by generating captions for a set of test-images which are scored via perceptual natural language processing metrics like BLEU (Elman, 1990) or CIDEr (Vedantam et al., 2015). These metrics assess similarity between the generated captions and the human-provided references. Despite large datasets, these models tend to over-fit to the training data, producing generic descriptions with word-repetitions, reciting dataset biases, and possessing a lack of qualitative diversity (Wang et al., 2017c; Aneja et al., 2018). Directly optimizing accuracy metrics via policy gradient further reduces diversity as the policy is encouraged to focus on a single mode.

More recently, generation of diverse image descriptions has received an increasing amount of attention (Deshpande et al., 2019; Shetty et al., 2017; Dai et al., 2017; Li et al., 2018; Aneja et al., 2019; Hu et al., 2020; Wang & Chan, 2019). For instance, latent variable models based on variational auto-encoders (VAEs), which represent sentence uncertainty through stochastic random variables, have been proven to be effective. However, these methods exhibit a noticeable trade-off between diversity and accuracy. Indeed, these approaches often only report the *oracle* CIDEr scores – a collection of captions is produced for an image via sampling in the learned latent space, the CIDEr for each is calculated and the highest value is reported. Such a procedure often masks the brittleness of learned models. For instance, in Figure 5.1(a), we show the different captions generated from a trained Seq-

| Image | Seq-CVAE | Ours |
|---|---|---|
|  | a woman sitting on a sidewalk with a suitcase (0.013) | a young boy is standing on the street with a fire hydrant (1.78) |
| | a young girl is sitting on a skateboard on the sidewalk (0.34) | a young girl is next to a person on a fire hydrant (2.28) |
| | a little girl is sitting on a red fire hydrant (1.66) | a young girl is standing on a fire hydrant (2.86) |

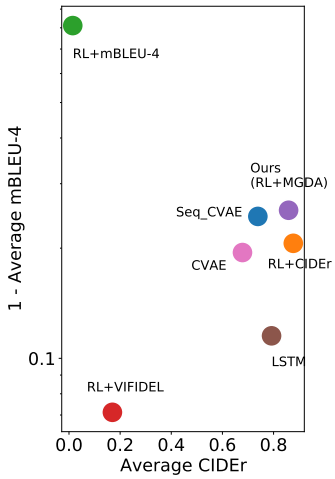



Figure 5.1: **(a)** In the left table, we show captions with worst, average and best CIDEr score from Seq-CVAE (Aneja et al., 2019) and our method. Our method outperforms Seq-CVAE significantly on the worst caption and performs on par on the best caption. **(b)** In the right plot, we show the performance of various models with respect to the perceptual ($x$-axis) and diversity ($y$-axis) metrics. Our method locates in the most upper-right position, demonstrating the effectiveness of the proposed approach in addressing multiple metrics.

CVAE model (Aneja et al., 2019) by Gaussian sampling in the learned latent space; the model parameters remain fixed. While the caption corresponding to the best CIDEr is good (in green), there is a degradation in the quality of the caption corresponding to the lowest CIDEr (in red). We contrast this with our method which significantly outperforms Seq-CVAE on the worst caption, and performs on par on the best caption. We show through our experiments that we learn robust models, in that they generate captions that are diverse and reliably accurate, on average and also worst-case behaviors.

Our goal is to bring the spotlight on the unaligned image captioning demands which require to achieve both accuracy and diversity at the same time. Adequately attending to both objectives remains a daunting task. As a step towards this goal, we propose to train models in a more balanced manner.

To achieve this, we propose to jointly optimize for accuracy metrics and a diversity measure (mBLEU-4), which estimates the mutual overlap between sentences. Naturally, the optimization dynamics are quite sensitive to the weighting of the various loss terms. Instead of hand-tuning the weights or employing heuristics, we frame the problem as a multi-objective optimization and use the multiple-gradient descent algorithm (MGDA) (Désidéri, 2012) to derive adaptive weights for the different loss terms. This permits to achieve Pareto optimality (Désidéri, 2012) in this multi-objective space. This is pictorially depicted in Fig. 5.1(b). $x$-axis and $y$-axis are indicators of accuracy (average-CIDEr) and diversity (average-

mBLEU4), respectively, and the goal is to be farther along *both* the axes. Unsurprisingly, the extremes on the $x$-axis and $y$-axis are occupied by reinforcement learning optimizing solely for CIDEr and mBLEU4, respectively. But our method locates in the most upper-right position, demonstrating the effectiveness of the proposed approach in simultaneously addressing multiple desired metrics.

We illustrate the efficacy of the proposed approach to generate good-quality as well as diverse captions on the COCO dataset (Lin et al., 2014). We also provide an exhaustive ablation study analyzing the effects of optimizing several accuracy and diversity metrics on LSTM and VAE-based models. Further analysis using radar plots reveals that our method makes the least compromise despite unaligned objectives, producing captions that are diverse while being reliably accurate on average.

## 5.2 LOSSES FOR DIVERSE IMAGE CAPTIONING

We are interested in generating a diverse set of captions $x^m$, $m \in \{1, \ldots, M\}$, given an image $I$. For readability we drop the index $m$ henceforth. Each generated caption $x = (x_1, \ldots, x_T)$ is a tuple of words $x_t \in \mathcal{X}$, $t \in \{1, \ldots, T\}$, each from a discrete vocabulary $\mathcal{X}$. Given an image $I$ we devise a probabilistic model $p_\theta(x|I)$ which depends on parameters $\theta$ and assigns a probability to every caption $x$.

To effectively sample from this probabilistic space we assume the probability distribution $p_\theta(x|I)$, jointly defined over all words $x_t$, $t \in \{1, \ldots, T\}$, of a caption, to factorize into a product of word-conditionals, *i.e.*,

$$p_\theta(x|I) = \prod_{t \in \{1,\ldots,T\}} p_\theta(x_t|x_{<t}, I). \tag{5.1}$$

This factorization enforces a temporal ordering, *i.e.*, the probability distribution for word $x_t$ is conditioned on all preceding words $x_{<t}$. Importantly, because the conditional's domain is the vocabulary space $\mathcal{X}$ and not a product space thereof, as it is the case for the joint distribution $p_\theta(x)$, ancestral sampling is a suitable and effective technique to generate a diverse set of captions.

In practice, the conditional probability distributions $p_\theta(x_t|x_{<t}, I)$, often also referred to as the decoder distributions, are modeled via recurrent LSTM nets or convolutional networks with masked filters. These networks aggregate the temporal history $x_{<t}$ into an efficient representation that is used to predict the current word $x_t$. However, given the preceding words $x_{<t}$, the conditional distribution $p_\theta(x_t|x_{<t}, I)$ has to cover many words which are

suitable to complete the sentence. This is important for the generation of diverse captions for any given image. While LSTM-nets can potentially model complex distributions, latent-variable models have been shown to be more effective at capturing the model uncertainty, due to the high representational capacity afforded by inclusion of stochastic latent variables.

Conditional Variational Auto Encoders (CVAE) and Sequential Conditional Variational Auto Encoders (Seq-CVAE) (Aneja et al., 2019) are recent VAE-based approaches that study recurrent latent-variable models for diversity in image captioning. Following typical variational inference techniques, both methods use an approximate posterior distribution over the latent variables, which is optimized by maximizing a lower bound (known as the Evidence Lower Bound, ELBo) to the maximum likelihood objective. CVAEs use a single latent variable $z$ for the entire caption produced from the approximated posterior $q_\phi(z|x, I)$. Consequently, the ELBo is given by:

$$\mathbb{E}_{z\sim q_\phi(z|x,I)}\Big[\sum_t (\log p_\theta(x_t|x_{<t}, z, I) - D_{KL}(q_\phi(z|x, I)||p(z)), \tag{5.2}$$

where $D_{KL}$ is the KL-divergence and $p(z)$ is a prior. The intuition behind the method is to capture sentence-level semantic diversity in the latent variables $z$, which is used to produce diverse captions by conditioning the decoder $p_\theta$ on $z$. In Seq-CVAE, the authors go one step further and introduce latent variables for each timestep $z_t$. The idea is to learn a rich representation at each timestep which captures the *intention* about the various ways a sentence can be completed from the current timestep on-wards. The approximate posterior is modified to $q_\phi(z_t|z_{t-1}, x, I)$, *i.e.*, it differs at each timestep and conditions on the latent variable from the previous timestep. The ELBo is given by:

$$\mathbb{E}_{z_1,\ldots,z_T\sim q_\phi(z_1,\ldots,z_T|x,I)}\Big[\sum_t \log p_\theta(x_t|x_{<t}, z_{\leq t}, I) - D_{KL}(q_\phi(z_t|z_{t-1}, x, I)||p_\theta(z_t|z_{<t}, x_{<t}, I))\Big]. \tag{5.3}$$

However, while remarkably diverse captions have been illustrated, it remains unclear how optimizing for diversity influences accuracy. We study the accuracy-diversity trade-offs in the following sections.

## 5.3 APPROACH

As discussed before, a variety of losses have been considered for diverse image captioning. Generally, those losses maximize a lower bound on the data log-likelihood. While this is

appealing to accurately capture the probability distribution, perceptual metrics are ignored.

To alleviate this, in the following, we discuss our approach to jointly optimize for both diversity and perceptual metrics. We achieve this by combining perceptual metrics with those that encourage diversity. Since those metrics are often non-differentiable, we use policy-gradient techniques as discussed in Section 5.3.1.

While prior work has used policy-gradient methods to optimize for multiple perceptual image captioning metrics, we found the combination of the losses/rewards to be mostly intuition based. For instance, Liu et al. (2017) advocate to add rewards by ensuring that their magnitudes are approximately balanced. While intuitive, training of those approaches requires a lot of strong priors about metrics's magnitudes on specific datasets. We address this concern by using the multiple-gradient descent algorithm (MGDA) as discussed in Section 5.3.2. The overall framework of our approach are depicted in Figure 5.2.

### 5.3.1 Policy Gradients and Reward Metrics

Complimentary to using the maximum likelihood loss for supervised learning with human-provided captions, reinforcement-learning (RL) algorithms have also been utilized for training image-captioning models (Ranzato et al., 2015; Rennie et al., 2017b; Dai et al., 2017; Liu et al., 2017). The key benefit of using RL is that model-free policy-gradient methods such as REINFORCE (Williams, 1992) enable the calculation of the gradients of the captioning model (the policy) even when the loss function is non-differentiable.

In RL, an agent interacts with an environment in a closed loop; at each timestep, the agent observes a *state* and takes an action according to its policy. It receives a scalar reward and the next state from the environment. This process continues until the end of an episode. The objective is to maximize the expected sum of rewards over the episode.

Sequence generation tasks such as image captioning closely follow this RL framework. Concretely, assume image captioning using an LSTM-based architecture similar to Karpathy & Fei-Fei (2015). At each timestep $t$, a distribution over the next word $x_t$ is obtained from $p_\theta(x_t|h_t, x_{t-1}, I)$, where $h_t$ is the LSTM hidden state encapsulating the sentence history $\{x_{<t-1}\}$, $x_{t-1}$ is the previous word, $I$ are the image features, and $\theta$ are the trainable parameters. The LSTM net can be viewed as a policy ($p_\theta$) which selects an action $x_t$ based on the current state $\{h_t, x_{t-1}, I\}$, resulting in a transition to the next state. The action-space for the policy is the size of the vocabulary $|\mathcal{X}|$. A non-zero environment reward is only provided at the end of the episode when an EOS (end-of-sentence) token is sampled. The policy receives 0 reward at all other timesteps.

The reward is generally computed by evaluating the generated caption against the ground-

truth captions, using metrics such as BLEU and CIDEr.

Denoting the sampled caption by $x = (x_1, \ldots, x_T)$, the RL objective to be maximized can be written as:

$$\eta(\theta) = \mathbb{E}_{x \sim p_\theta}[r(x)], \tag{5.4}$$

where $r(x)$ is the reward for the sequence $x$ and $p_\theta(x)$ is the probability that the policy generates the caption $x$.

**Policy Gradients with REINFORCE.** REINFORCE (Williams, 1992), also known as the score function estimator, is applicable for maximizing the expected reward specified in Eq. (5.4). It enables the conversion of a gradient of the expectation into an expectation over the weighted gradient of the score function $\nabla_\theta \log p_\theta(x)$. Concretely,

$$\nabla_\theta \eta(\theta) = \mathbb{E}_{x \sim p_\theta}[r(x) \nabla_\theta \log p_\theta(x)]. \tag{5.5}$$

We obtain an unbiased estimate of the gradient using Monte Carlo sampling. Let $\{x^1, \ldots, x^k\}$ be $k$ captions generated using the policy $\theta$. Using the auto-regressive model for $p_\theta(x)$ defined in Section 5.2 and denoting the state $\{h_t, x_{t-1}, I\}$ at any timestep with $s_t$, we obtain the gradient via

$$\nabla_\theta \eta(\theta) = \sum_{i=1}^{k} \sum_{t=1}^{T} \left[ r(x^{(i)}) \nabla_\theta \log p_\theta(x_t^{(i)} | s_t^{(i)}) \right]. \tag{5.6}$$

Intuitively, the gradients increase (reinforce) the probability of producing the words (or actions) which result in high reward $r(x)$ for the final caption, as obtained using the standard evaluation metrics. To reduce the variance in the policy gradient estimates, control variate methods are used. A common approach is to incorporate a baseline function $b$ which is independent of the sampled random variables $x$ by replacing $r(x)$ in the above equation with $r(x) - b$. Finally, for readability we rewrite the RL objective as a loss function to be minimized. We also make explicit the dependence of the loss on the chosen reward metric $r$. Therefore, in summary, we target the loss:

$$L_{\mathrm{RL}}(\theta; r) = -\eta(\theta) = -\mathbb{E}_{x \sim p_\theta}[r(x)]. \tag{5.7}$$

### 5.3.2 RL Optimization using MGDA

As illustrated in Section 5.3.1, sequential models such as image-caption generators are easily optimized with common RL methods using the loss $L_{\mathrm{RL}}(\theta; r)$. $L_{\mathrm{RL}}(\theta; r)$ increases the

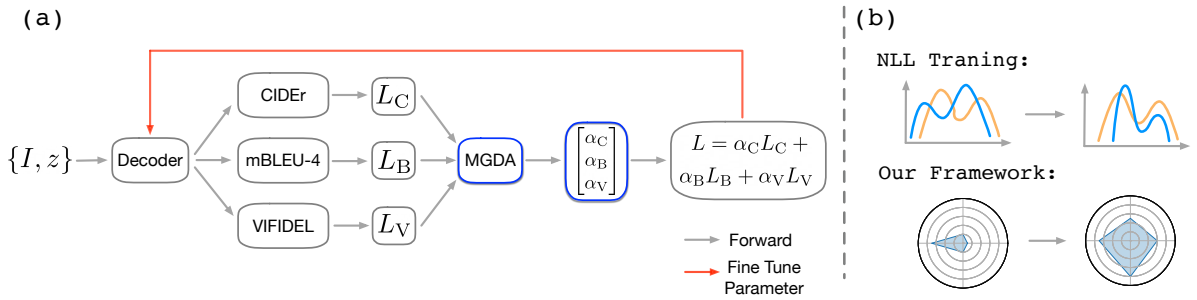



Figure 5.2: **Overview of our approach.** **(a)** illustrates the pipeline of our proposed approach. The VAE decoder is pre-trained by maximizing the ELBo. We use the RL-based multi objective optimization to fine tune the decoder with gradients from $L_c$, $L_B$ and $L_V$, which correspond to loss terms CIDEr, mBLEU-4 and VIFIDEL respectively. $\alpha_C$, $\alpha_B$ and $\alpha_V$ are the weights learned by the MGDA approach. On the right, **(b)** shows that the negative log-likelihood (NLL) pre-training tries to fit data distribution. The MGDA based optimization then moves this learned distribution to the region of high balance of diversity and accuracy.

probability of sampling captions $x$ which have a large reward $r(x)$.

The choice of reward function is important. Language evaluation metrics such as CIDEr, BLEU, and SPICE have been used in prior work as reward functions (Ranzato et al., 2015; Rennie et al., 2017b). Pseudo reward functions learnt with adversarial training have also been proposed (Dai et al., 2017). Furthermore, it is possible to combine the different metrics for joint optimization. For instance, in (Liu et al., 2017), the authors use a linearly weighted sum of the different metrics as the reward in a policy gradient algorithm; the weight on each metric is manually chosen.

We argue that hand-tuning the contribution of the various metrics in the overall reward function is a difficult proposition, leads to sub-optimal regions of the policy space if done incorrectly and requires a lot of manual tuning. This follows from the widely accepted premise that deep-RL algorithms are quite sensitive to the design of the reward functions.

The issue of manually selecting weights is further exacerbated when the metrics are either competing or exhibit a trade-off. For instance, given our objective of generating diverse captions, we propose to use m-BLEU4 (detailed in Section 5.4.3) as one of the rewards. However, when m-BLEU4 is used in conjunction with a CIDEr reward, which promotes caption accuracy and therefore hurts the performance of diversity (Luo & Shakhnarovich, 2020), apportioning of weights is not obvious to these competing rewards.

To deal with this complexity, we propose to frame the problem of learning the model (policy) with different RL rewards as a *multi-objective optimization*, with the following vector

valued loss function:

$$\min_{\theta} \mathbf{L}_{\mathrm{RL}}(\theta) = \min_{\theta} (L_{\mathrm{RL}}(\theta; r^1), \ldots, L_{\mathrm{RL}}(\theta; r^k))^T, \tag{5.8}$$

where $(r^1, \ldots, r^k)$ are the different reward metrics. Unlike the single-objective scenario, an update algorithm for multi-objective optimization aims to achieve the Pareto optimality, which in our context is defined as follows:

1. A solution $\theta$ dominates a solution $\tilde{\theta}$ if $L_{\mathrm{RL}}(\theta; r^i) \leq L_{\mathrm{RL}}(\tilde{\theta}; r^i)$, $\forall$ reward metrics $i \in \{1, \ldots, k\}$, and $\mathbf{L}_{\mathrm{RL}}(\theta) \neq \mathbf{L}_{\mathrm{RL}}(\tilde{\theta})$.

2. A solution $\theta^*$ is called Pareto optimal if there exists no solution $\theta$ that dominates $\theta^*$.

The Multiple Gradient Descent Algorithm (MGDA) (Désidéri, 2012) is an efficient gradient-based method to solve the multi-objective optimization problem to local optimality, also known as a Pareto stationary point. Following the ideas in (Désidéri, 2012), it can be shown that the gradient direction along which all the losses $L_{\mathrm{RL}}(\theta; r^i)$ are reduced is given by the solution to the following norm-minimization problem:

$$\min_{\alpha_i, \ldots, \alpha_k} \left\{ \left\| \sum_{i=1}^{k} \alpha_i \nabla_\theta L_{\mathrm{RL}}(\theta; r^i) \right\|_2^2 \;\middle|\; \sum_{i=1}^{k} \alpha_i = 1, \alpha_i \geq 0, \; \forall i \right\}. \tag{5.9}$$

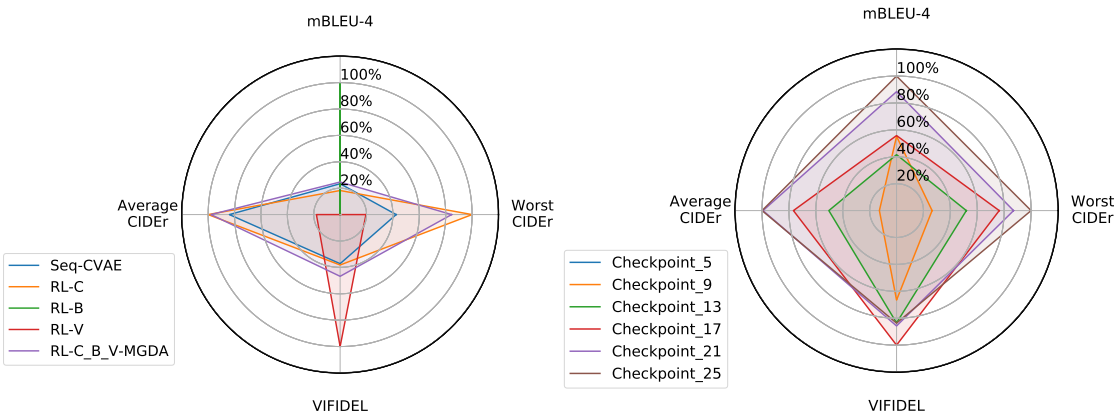
Sener & Koltun (2018) show that this constrained optimization can be solved by using the Frank-Wolfe algorithm, yielding the *argmin* values $\{\bar{\alpha}_i\}_1^k$ which can then be used to perform gradient descent on the policy parameters:

$$\theta \leftarrow \theta - \eta \sum_{i=1}^{k} \bar{\alpha}_i \nabla_\theta L_{\mathrm{RL}}(\theta; r^i). \tag{5.10}$$

The overall algorithm involves alternating between solving Eq. (5.9) to find the weight coefficients and updating the model parameters using Eq. (5.10).

## 5.4 RESULTS

In the following, we first describe the experimental setting, followed by our results and analysis.




(a) Across models (b) Across training epochs

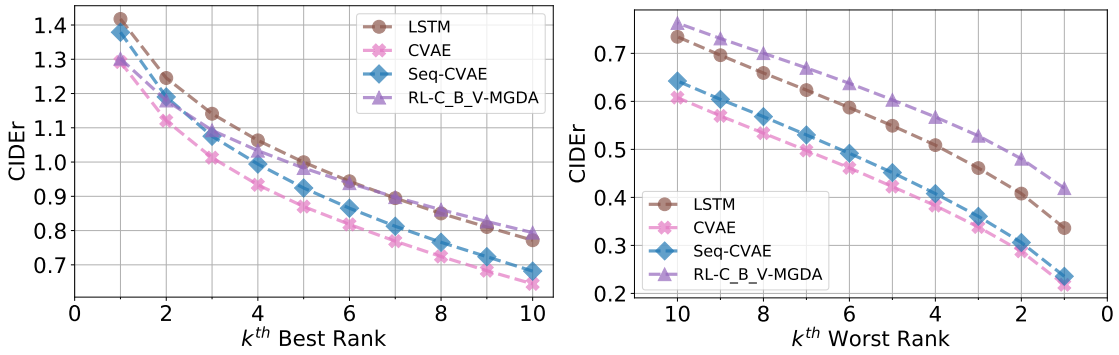
Figure 5.3: **Radar plots of normalized performance.** Our approach is most balanced. For each metric, we have normalized the performance as the percentage with respect to the best performance we could have from all models. Namely, best performance is marked as 100% while worst performance is represented as 0%. **(a)** We use baseline Seq-CVAE (Aneja et al., 2019) and perform ablations of our approach on Aneja et al. (2019). Models optimized with single metric perform the best on that metric's axis while suffering miserably along all the other axes. For example, `RL-B` yields huge improvement on mBLEU-4 while faring poorly on the perceptual metrics. Because `RL-B` performs worst on all metrics other than mBLEU-4, it reaches 100% on mBLEU-4 while 0% on the other metrics. On the contrary, our model has the most balanced performance. It improves the average performance compared to vanilla Seq-CVAE. **(b)** We shows the evolution of our approach as training progresses. This is done on the val split of Karpathy & Fei-Fei (2015). Clearly our model oscillates between optimising for different metrics, finally converging to performing in a balanced way along all the metrics. Note that checkpoint 5 of RL fine-tuning performs worst on every metric hence it degenerates to a point at the center. (Best viewed in color.)

### 5.4.1 Experimental Settings

**Dataset:** We use the COCO dataset (Lin et al., 2014) for our experiments, specifically, we use the split provided by Karpathy & Fei-Fei (2015). It has 113,287 train, 5,000 val and 5,000 test images. For each image, there are 5 reference captions.

**Baselines:** We use the following approaches for our experiments as baselines:

- **LSTM:** the standard single decoder structure `LSTM`.
- **Conditional Variational Autoencoder:** for encoder–decoder style models, we study `CVAE (z@t=0)` and `CVAE (z@t=0-T)`, each of which is a conditional variational auto–encoder with a *unidirectional* encoder. `CVAE (z@t=0)` uses latent variable $z$ in the decoder only at $t = 0$ while `CVAE (z@t=0-T)` uses the same $z$ at every timestep of decoding. We also study two more baselines by instead utilizing a *bidirectional* encoder in `CVAE (z@t=0)` and `CVAE (z@t=0-T)` through ELMo (Peters et al., 2018).




(a) Top-10 captions (b) Bottom-10 captions

Figure 5.4: **Top-k and Bottom-k Accuracy. (a)** shows the CIDEr score of the top-10 captions. Our model performs on par with the other approaches of LSTM and CVAE and also with respect to the baseline Seq-CVAE. Note that although the performance on the top-1 caption is lower than the best model, there is crossover point making our approach have an edge on average. A better average performance is desirable for a diverse captioning model. **(b)** shows the bottom-10, *i.e.* the worst captions. Our approach significantly surpasses all the other methods shown. Note, we use beam search to generate 20 captions, per sampled latent vector, $z$. We do this for 20 $z$ per image. (This plot is best viewed in color).

• **Sequential Conditional Variational Autoencoder:** the recently proposed Seq-CVAE (Aneja et al., 2019) is another baseline. We provide results on this baseline systematically by first only individually optimizing for accuracy (CIDEr) `RL-C`, diversity (m-BLEU4) `RL-B`, and relevance (VIFIDEL) `RL-V`. `RL-C_B-MGDA` refers to MGDA optimization for CIDEr and m-BLEU4. We refer to our method as `RL-C_B_V-MGDA`. It jointly optimizes for all three metrics with MGDA as mentioned in Section 5.3.2. Lastly `RL-C_B_V-Heuristic` uniformly weights the three losses.

**Implementation details:** For all models, we utilize a 512-dimensional embedding of the image features obtained from the fc7 layer of a VGG16 network (Simonyan & Zisserman, 2014) that is pre-trained on the ImageNet dataset. The word embedding is also 512-dimensional like the dimension of the latent variable $z$ if used. Motivated from the KL cost annealing mentioned by Bowman et al. (2016), we use as weight for the KL term in the ELBo objective (Eq. (5.2) and Eq. (5.3)) 0.01 for all `CVAE` baselines and 0.1 for all `Seq-CVAE (z@t=0-T)` based models. We use Adam (Kingma & Ba, 2015) with a learning rate of $5e^{-4}$ to train the `LSTM` or to optimize the ELBo of all VAE-like models. To fine tune `Seq-CVAE (z@t=0-T)` after ELBo training, we use Adam with a learning rate of $5e^{-5}$ to maximize the set of rewards.

| | Models | ELMo | RL-C | RL-B | RL-V | 1. Worst CIDEr ↑ | 2. Average CIDEr ↑ | 3. Oracle CIDEr ↑ | 4. VIFIDEL ↑ | 5. Novel Caps(%) ↑ | 6. Distinct Caps(%) ↑ | 7. mBLEU-4 ↓ | 8. Div-2 ↑ | 9. Div-4 ↑ |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 1 | LSTM | - | | | | $0.34_{\pm 0.36}$ | $0.79_{\pm 0.56}$ | $1.42_{\pm 0.87}$ | $0.48_{\pm 0.11}$ | 60.44 | $100.0_{\pm 0.00}$ | $0.88_{\pm 0.05}$ | $0.19_{\pm 0.03}$ | $0.27_{\pm 0.05}$ |
| 2 | CVAE (z@t=0) | | | | | $0.28_{\pm 0.31}$ | $0.68_{\pm 0.48}$ | $1.18_{\pm 0.73}$ | $0.49_{\pm 0.12}$ | 86.69 | $70.87_{\pm 12.0}$ | $0.81_{\pm 0.07}$ | $0.27_{\pm 0.05}$ | $0.35_{\pm 0.06}$ |
| 3 | CVAE (z@t=0-T) | | | | | $0.22_{\pm 0.25}$ | $0.63_{\pm 0.42}$ | $1.19_{\pm 0.69}$ | $0.49_{\pm 0.11}$ | 90.47 | $94.15_{\pm 6.30}$ | $0.74_{\pm 0.08}$ | $0.28_{\pm 0.05}$ | $0.42_{\pm 0.06}$ |
| 4 | CVAE (z@t=0) | ✓ | | | | $0.23_{\pm 0.27}$ | $0.64_{\pm 0.45}$ | $1.18_{\pm 0.71}$ | $0.49_{\pm 0.12}$ | 88.15 | $83.77_{\pm 12.4}$ | $0.80_{\pm 0.07}$ | $0.26_{\pm 0.06}$ | $0.35_{\pm 0.07}$ |
| 5 | CVAE (z@t=0-T) | ✓ | | | | $0.27_{\pm 0.31}$ | $0.61_{\pm 0.45}$ | $1.05_{\pm 0.67}$ | $0.49_{\pm 0.12}$ | 90.23 | $71.68_{\pm 15.6}$ | $0.84_{\pm 0.06}$ | $0.23_{\pm 0.05}$ | $0.32_{\pm 0.06}$ |
| 6 | Seq-CVAE (z@t=0-T) | ✓ | | | | $0.33_{\pm 0.36}$ | $0.74_{\pm 0.52}$ | $1.29_{\pm 0.78}$ | $0.49_{\pm 0.11}$ | 78.02 | $72.97_{\pm 17.6}$ | $0.76_{\pm 0.08}$ | $0.29_{\pm 0.06}$ | $0.40_{\pm 0.07}$ |
| 7 | RL-C | ✓ | ✓ | | | $0.77_{\pm 0.68}$ | $0.88_{\pm 0.68}$ | $0.99_{\pm 0.73}$ | $0.49_{\pm 0.12}$ | 68.10 | $12.23_{\pm 8.97}$ | $0.79_{\pm 0.23}$ | $0.64_{\pm 0.22}$ | $0.54_{\pm 0.13}$ |
| 8 | RL-B | ✓ | | ✓ | | $0.00_{\pm 0.01}$ | $0.01_{\pm 0.03}$ | $0.06_{\pm 0.08}$ | $0.36_{\pm 0.09}$ | 100.0 | $46.20_{\pm 27.3}$ | $0.19_{\pm 0.22}$ | $0.70_{\pm 0.18}$ | $0.68_{\pm 0.16}$ |
| 9 | RL-V | ✓ | | | ✓ | $0.15_{\pm 0.19}$ | $0.17_{\pm 0.20}$ | $0.19_{\pm 0.22}$ | $0.71_{\pm 0.21}$ | 100.0 | $15.20_{\pm 13.5}$ | $0.93_{\pm 0.10}$ | $0.24_{\pm 0.13}$ | $0.30_{\pm 0.13}$ |
| 10 | RL-C_B-MGDA | ✓ | ✓ | ✓ | | $0.64_{\pm 0.60}$ | $0.86_{\pm 0.65}$ | $1.12_{\pm 0.76}$ | $0.49_{\pm 0.12}$ | 78.39 | $28.70_{\pm 17.0}$ | $0.75_{\pm 0.16}$ | $0.42_{\pm 0.17}$ | $0.43_{\pm 0.11}$ |
| 11 | RL-C_B_V-MGDA | ✓ | ✓ | ✓ | ✓ | $0.65_{\pm 0.60}$ | $0.86_{\pm 0.64}$ | $1.10_{\pm 0.75}$ | $0.52_{\pm 0.13}$ | 83.21 | $29.60_{\pm 16.7}$ | $0.75_{\pm 0.16}$ | $0.40_{\pm 0.17}$ | $0.43_{\pm 0.10}$ |
| 12 | RL-C_B_V-Heuristic | ✓ | ✓ | ✓ | ✓ | $0.66_{\pm 0.61}$ | $0.87_{\pm 0.65}$ | $1.11_{\pm 0.76}$ | $0.52_{\pm 0.13}$ | 81.96 | $28.00_{\pm 16.0}$ | $0.74_{\pm 0.17}$ | $0.42_{\pm 0.17}$ | $0.44_{\pm 0.11}$ |

Table 5.1: **Perceptual and Diversity Evaluation.** We use 20 samples. The samples are generated either using beam search of beam size of 20 (LSTM) or by sampling 20 latent vectors ($z$) from the latent space (VAE). The results are reported as `mean`$\pm$`std`. The `mean` is taken over the diverse captions for a given image and then averaged over all the images. The `std` is evaluated over all the captions for all the images. This `std` demonstrates each model's robustness over different metrics. **Green** indicates best, while **Red** indicates worst. Reported on Test split of Karpathy & Fei-Fei (2015).(Best viewed in color)

### 5.4.2 Striving for a Balanced Objective

Captioning methods ideally generate a variety of descriptions without losing relevance and grammatical robustness. In Figure 5.3(a) we observe the proposed approach to perform well on both **diversity** and **accuracy** metrics.

Baselines often perform well on either diversity (mBLEU-4) or on accuracy (Average CIDEr) but not on both. The method encapsulating larger area with higher numbers on all axes is the most desirable method for diverse image captioning. Note in Figure 5.3(a), our approach has even spread along all axes.

Further, Figure 5.3(b) shows the progression of this enveloped area as the training progresses. The uneven stretch along different axes during training, that ends in a balanced version indicates that the model begins by independently optimising individual metrics, but eventually converging to the weighting for best joint optimization along all of them.

We sample multiple captions from each of the models and sort them (in decreasing order) based on their CIDEr scores. Worst CIDEr corresponds to the score achieved by the caption at the bottom of this ranking. A low worst CIDEr score typically indicates that at least a few of the captions sampled from the model can not be accepted as reasonable descriptions for the image. In Figure 5.4(b) we note that the worst CIDEr from our model is much higher than the worst CIDEr from the baseline methods. Additionally, in Figure 5.4(b) we show the CIDEr performance of the ten top ranking captions, where our approach performs on par with baselines, indicating that our approach does not harm baseline's performance.

**Evaluation Criteria.** We compare all aforementioned approaches on the following accuracy and diversity in Table 5.1:

•**Accuracy.** In Section 5.4.3, we report the Top-1 accuracy, Average Accuracy, Worst

| | Approach | CIDEr | Meteor | ROUGE-L | BLEU-1 | BLEU-2 | BLEU-3 | BLEU-4 |
|---|---|---|---|---|---|---|---|---|
| Worst | Uniform | $0.32_{\pm 0.32}$ | $0.16_{\pm 0.07}$ | $0.35_{\pm 0.11}$ | $0.44_{\pm 0.13}$ | $0.20_{\pm 0.17}$ | $0.06_{\pm 0.13}$ | $0.02_{\pm 0.08}$ |
| | BMCR | $0.35_{\pm 0.35}$ | $0.16_{\pm 0.07}$ | $0.35_{\pm 0.11}$ | $0.46_{\pm 0.14}$ | $0.22_{\pm 0.18}$ | $0.06_{\pm 0.13}$ | $0.01_{\pm 0.07}$ |
| | Ours | $0.34_{\pm 0.36}$ | $0.15_{\pm 0.07}$ | $0.34_{\pm 0.11}$ | $0.46_{\pm 0.14}$ | $0.21_{\pm 0.18}$ | $0.05_{\pm 0.13}$ | $0.01_{\pm 0.06}$ |
| Average | Uniform | $0.71_{\pm 0.51}$ | $0.23_{\pm 0.09}$ | $0.47_{\pm 0.13}$ | $0.60_{\pm 0.14}$ | $0.39_{\pm 0.19}$ | $0.22_{\pm 0.19}$ | $0.12_{\pm 0.17}$ |
| | BMCR | $0.78_{\pm 0.55}$ | $0.24_{\pm 0.09}$ | $0.49_{\pm 0.13}$ | $0.64_{\pm 0.15}$ | $0.44_{\pm 0.19}$ | $0.26_{\pm 0.20}$ | $0.15_{\pm 0.17}$ |
| | Ours | $0.77_{\pm 0.55}$ | $0.24_{\pm 0.09}$ | $0.49_{\pm 0.13}$ | $0.64_{\pm 0.15}$ | $0.44_{\pm 0.19}$ | $0.26_{\pm 0.20}$ | $0.14_{\pm 0.17}$ |
| Oracle | Uniform | $1.28_{\pm 0.81}$ | $0.35_{\pm 0.17}$ | $0.63_{\pm 0.17}$ | $0.78_{\pm 0.16}$ | $0.62_{\pm 0.22}$ | $0.47_{\pm 0.27}$ | $0.33_{\pm 0.31}$ |
| | BMCR | $1.38_{\pm 0.86}$ | $0.37_{\pm 0.19}$ | $0.65_{\pm 0.17}$ | $0.82_{\pm 0.15}$ | $0.68_{\pm 0.21}$ | $0.54_{\pm 0.27}$ | $0.41_{\pm 0.32}$ |
| | Ours | $1.36_{\pm 0.86}$ | $0.36_{\pm 0.18}$ | $0.65_{\pm 0.17}$ | $0.83_{\pm 0.15}$ | $0.68_{\pm 0.21}$ | $0.55_{\pm 0.28}$ | $0.40_{\pm 0.33}$ |

Table 5.2: **Comparison between heuristic weights and MGDA.** We compare the performance between BMCR discussed in Liu et al. (2017) and ours. We report the evaluation in the form $\texttt{mean}_{\pm\texttt{std}}$ across the test set in Karpathy & Fei-Fei (2015). With no requirement of prior knowledge about the metrics on COCO, our approach behaves similar to BMCR, emphasizing our effectiveness.

Performance evaluated on CIDEr. The fidelity of the generated caption to the image is evaluated using VIFIDEL.

- **Diversity.** Our evaluation of diversity performance is presented in Section 5.4.3.

Note, the on-par performances between our MGDA-based joint optimization approach `RL-C_B_V-MGDA` (Row 11) and heuristic approach with uniform weights `RL-C_B_V-Heuristic` (Row 12) are mainly due to similar magnitudes of metrics for rewards, *i.e.* CIDEr, mBLEU-4, and VIFIDEL. We further demonstrate the effectiveness of MGDA that no prior knowledge is required in Section 5.4.4.

### 5.4.3 Accuracy

**Worst and Average CIDEr (Col 1 and 2).** Contrary to recent approaches, in Table 5.1 left part we report the worst and average CIDEr obtained by all models. This resembles a worst-case analysis of a model. Since our goal is to generate multiple captions from a captioning model, we think a high worst CIDEr score combined with a high average score is more useful, than a high top-1 accuracy. The motivation is that evaluating each generated caption is infeasible, a model trained to generate multiple captions should have high accuracy on average. Within expectation, from Table 5.1 we observe that optimization for CIDEr (Row 7) performs best in the worst and average CIDEr. However this approach performs worst on almost all diversity metrics (Table 5.1 right part, discussed in Section 5.4.3). Our MGDA based approach (Row 11) is more balanced and results in a decent CIDEr score while not suffering on diversity metrics.

**Oracle CIDEr (Col 3).** Following the approach of Wang et al. (2017d); Deshpande et al.

| Image | LSTM | CVAE | Seq-CVAE | Ours |
|---|---|---|---|---|
|  | a red stop sign on a city street at night (0.082) | a street sign with a street sign (0.00) | a street sign on a street corner at night (0.0020) | a traffic light at night with a street sign (1.34) |
| | a stop sign sitting on the side of a road (0.63) | a street light with a street sign (0.50) | a stop sign in front of a tall building (0.14) | a traffic light next to a street at night (1.44) |
| | a red stop light sitting on the side of a road (1.09) | a street light at night with traffic lights (0.79) | a street sign with a light on top of it (0.74) | a traffic light sitting on the side of a street (2.15) |
|  | a picture of a sandwich on a plate (0.50) | two plates of food with a knife and some food (0.52) | a plate of food with a fork and a fork (0.58) | a sandwich and a bowl of soup on a table (0.10) |
| | a close up of a sandwich on a plate with a spoon (0.95) | two plates of food on a table with food (0.70) | a plate of food with a fork and a spoon (0.69) | a white plate of a sandwich and a table (0.53) |
| | a close up of a plate of food on a table (2.11) | a plate of food with a knife and some food (1.04) | a plate of food on a wooden table (1.36) | a white plate topped with a sandwich and a knife (0.84) |

Table 5.3: Qualitative analysis. For the top image, our approach outperforms other models. In the bottom image, we show one case that our framework fails to improve upon baselines. The number following each caption is the corresponding CIDEr score. (Best viewed in color)

(2019); Aneja et al. (2019), we also report the top-1 accuracy or the oracle CIDEr. Although, Oracle metrics indicate the best possible performance of a model, they provide a limited signal regarding the quality of the multiple generated captions. Indeed, `LSTM` with beam search (Table 5.1's Row 1) outperforms all methods on Oracle CIDEr metric while producing the least diverse captions that are novel, *i.e.* performing worst in Col 5.

**VIFIDEL (Col 4).** We also report performance on VIFIDEL (Madhyastha et al., 2019), a recently proposed metric that measures 'faithfulness' of the caption to image content. It assesses the similarity of the objects in the image to the words in the generated caption. From Row 9 of Table 5.1 we see: optimizing for only VIFIDEL leads to a sharp decline in all the other perceptual metrics. However, after adding VIFIDEL to our proposed objective, we end up improving the *worst* performance on the CIDEr metric (Row 11 *vs*. 10 on Col 1). Additionally, we show in the next section that this joint optimization increases the diversity of the generated captions.

Evaluation of Diversity **Novel Captions (Col 5).** A caption generated during evaluation is tagged as *novel* if the same caption is not present in the training corpus. This measure therefore indicates the model's ability to learn concepts and combine them for better generalization. Our `RL-C_B_V-MGDA` approach (Row 11 in Table 5.1) produces $\sim 83\%$ novel captions. Although `RL-B` and `RL-V` produce 100% novel captions, their performance on the perceptual

metrics is poor (see Table 5.1's Row 11 *vs*. 8 and Row 11 *vs*. 9 on Col 1, 2, and 3), suggesting that many of their novel captions are unfit descriptions of the image.

**Distinct Captions (Col 6).** This metric refers to the percentage of distinct captions generated by the model, for each image. Two captions qualify as *distinct* if they differ by at least one word position. For VAE based models, we generate 20 captions by sampling 20 times in the latent ($z$) space. For deterministic LSTM-based models, we run beam search with size 20. In Table 5.1, we observe that our method produces a lower percentage of distinct captions compared to baseline approaches (Row 11 *vs*. 8 on Col 6). However, the higher perceptual metrics, *e.g*. Average CIDEr, for our method suggests that this moderate number of generated captions are all quite relevant to the image and different from one another, which is not the case for the baselines (Row 11 *vs*. 8 on Col 1-4).

**Mutual Overlap, mBLEU-4 (Col 7).** mBLEU-4 measures the 4-gram overlap between the $M$ generated captions. For each of these $M$ captions, the BLEU-4 score is first calculated w.r.t. the remaining $M-1$ captions. This is then averaged across all captions and all test images to obtain the final mBLEU-4 score. A lower value of this score implies less mutual overlap between generated captions, and hence higher diversity. Note, in Table 5.1 our method achieves a lower mBLEU-4 of 0.74 compared to baselines that only optimize for perceptual accuracy (Row 11 *vs*. 7 on Col 7), demonstrating the importance of jointly optimizing competitive metrics.

**Diverse n-grams, Div-$n$ (Col 8 and 9).** Div-$n$ is calculated as the ratio of the number of unique $n$-grams to the total number of unique words for a set of $M$ generated captions. A higher value is better. It demonstrates that the model has learned useful representations that enable it to output varied word combinations or phrases. As expected, `RL-B` scores highest on this metric, but its word combinations are potentially not very useful since it performs poorly on perceptual metrics (Row 11 *vs*. 8 on Col 1-4).

### 5.4.4 Heuristic *vs*. MGDA

To demonstrate the effectiveness of our approach, in Table 5.2 we jointly optimize seven rewards, namely BLEU-{1, 2, 3, 4} (Papineni et al., 2001), METEOR (Banerjee & Lavie, 2005), CIDEr (Vedantam et al., 2015) and ROUGE-L (Lin & Hovy, 2003). For choosing weights heuristically, we use two approaches: 1) weigh each reward uniformly (Approach Uniform); 2) use weights from BMCR mentioned by Liu et al. (2017) which were chosen with strong prior knowledge about the scale of each metric on the COCO dataset (Approach BMCR). As can be seen, without strong prior knowledge as BMCR, Approach Uniform yields inferior performance. Our approach behaves similar to BMCR, without any manual

selection of weights and any prior information on any specific data.

## 5.5 RELATED WORK

**Image Captioning:** Generating captions and paragraphs for images has garnered significant interest over the last few years. In the 2000s, sentence-retrieval-based approaches were studied (Barnard et al., 2003), but were computationally expensive and difficult to generalize to a broad set of test cases. These issues have been somewhat mitigated with the use of deep-learning based approaches, which have demonstrated great success on challenging captioning datasets, *e.g.* MSCOCO (Lin et al., 2014).

To manage the high-dimensional space of captions, which grows exponentially with the number of words in a caption, auto-regressive decomposition of the joint probability space of words is effective (Karpathy & Fei-Fei, 2015; Vinyals et al., 2015b; Cho et al., 2014). This enables the use of recurrent neural nets (RNNs) and variants like long-short-term-memory (LSTM) (Hochreiter & Schmidhuber, 1997a) for sequential word generation. Image features extracted from a deep convolution network are also used within LSTMs and the model is trained with a maximum likelihood loss. These architectures have been extended in a plethora of ways, *e.g.*, including an object detector output as auxiliary information (Yao et al., 2017b), using spatial and temporal attention over the visual input (Xu et al., 2015; Anderson et al., 2018), and replacing recurrent nets with masked convolutions (Aneja et al., 2018). Evaluation of the image captioning models on the test-set images is typically done using metrics such as BLEU (Elman, 1990) and CIDEr (Vedantam et al., 2015). Recently, the VIFIDEL metric (Madhyastha et al., 2019) was proposed, which takes into account the semantic similarity between labels of objects depicted in images and words in the generated caption.

However, arguably, captions for an image are ambiguous: for a given image, many captions with different language characteristics are suitable, and it is desirable for a model to capture this diversity. Consequently, diverse image captioning is a major theme in recent works, which we briefly discuss next.

**Diverse Image Captioning:** Several techniques have been investigated to produce diverse captions. Beam search and Diverse beam search (Vijayakumar et al., 2018) are methods that, rather than greedily selecting the word with highest probability at each timestep, explore multiple sentence trajectories with the objective to sample from different modes of the sentence distribution. These are inference-only methods to improve diversity and require no model changes.

In PosCap (Deshpande et al., 2019), the authors show that diversity in the sentence struc-

ture (syntactic diversity) can be improved by using the part-of-speech tags in the captioning model. Prior work has also explored caption generators based on probabilistic generative models such as Generative Adversarial Networks (GAN) (Goodfellow et al., 2014a) and Variational Autoencoders (VAE) (Kingma & Welling, 2014; Pu et al., 2016). GAN based models attempt to match the distribution of captions produced by the generator with the distribution of the ground truth captions (Shetty et al., 2017; Dai et al., 2017; Li et al., 2018). While they have had success in diversifying the captions, they tend to lag behind likelihood based methods in terms of perceptual metrics (BLEU, CIDEr).

Among likelihood-based VAE approaches, AG-CVAE (Wang et al., 2017c) and Seq-CVAE (Aneja et al., 2019) are recent methods with a focus on diversity. AG-CVAE models the VAE prior with rich distributions and mimics semantic diversity by using the latent space to condition on different regions of the image. Seq-CVAE proposes a captioning network based on sequential VAEs, modeling the future utterances of the sentence in its per-word latent space. Since VAE based methods have been shown to provide a good balance between diversity and score on the perceptual metrics, we use them to build our model in this work.

However, different from prior work, we propose direct optimization of a non-differentiable diversity metric using reinforcement learning.

**Policy Gradient Methods for Captioning:** Reinforcement Learning (RL) algorithms, particularly policy gradient methods, have been used to train sequence generation models by optimizing sequence-based test metrics (Ranzato et al., 2015; Rennie et al., 2017b; Dai et al., 2017; Liu et al., 2017). Since the reward function can be non-differentiable, perceptual metrics have been directly maximized at training time. For instance, Ranzato et al. (2015) optimizes for BLEU, whereas Liu et al. (2017) manually designs a weighted combination of different metrics. Rennie et al. (2017b) show the importance of using a control variate for reduction of variance in the REINFORCE (Williams, 1992) algorithm. This is achieved using the inference-time rewards from the current model. Finally, Dai et al. (2017) utilizes policy gradient with proxy rewards obtained from an evaluator network, that is iteratively trained with the policy. One major shortcoming of these approaches is that the combination of rewards to be optimized with RL is mostly intuition-based, static, and requires a lot of hand-tuning from a domain expert.

We address this concern and propose an algorithm that provides dynamic, auto-tuned weighting between possibly competing reward objectives.

## 5.6 CONCLUSION

We study a multi-gradient objective optimization approach to directly optimize both perceptual and diversity metrics for image captioning via policy gradient. This method permits to strive for Pareto optimality of two competing goals: high accuracy and high diversity. The employed models exhibit encouraging results across both perceptual metrics and diversity metrics which is challenging to achieve with either classical likelihood based optimization or policy gradient on a reward composed of heuristically combined metrics of interest.

# Part III

# Tackling Prior and Aggregate-Posterior Mismatch in Variational Autoencoders

# CHAPTER 6: A CONTRASTIVE LEARNING APPROACH FOR TRAINING VARIATIONAL AUTOENCODER PRIORS

## 6.1 INTRODUCTION

Variational autoencoders (VAEs) (Kingma & Welling, 2014; Rezende et al., 2014) are one of the powerful likelihood-based generative models that have applications in image generation (Brock et al., 2018; Karras et al., 2019; Razavi et al., 2019), music synthesis (Dhariwal et al., 2020), speech generation (Oord et al., 2016; Ping et al., 2020), image captioning (Aneja et al., 2019; Deshpande et al., 2019; Aneja et al., 2018), semi-supervised learning (Izmailov et al., 2020), and representation learning (Van Den Oord et al., 2017; Fortuin et al., 2018).

Although there has been tremendous progress in improving the expressivity of the approximate posterior, several studies have observed that VAE priors fail to match the *aggregate (approximate) posterior* (Rosca et al., 2018; Hoffman & Johnson, 2016). This phenomenon is sometimes described as *holes in the prior*, referring to regions in the latent space that are not decoded to data-like samples. Such regions often have a high density under the prior but have a low density under the aggregate approximate posterior. The prior hole problem is commonly tackled by increasing the flexibility of the prior via hierarchical priors (Klushyn et al., 2019), autoregressive models (Gulrajani et al., 2016), a mixture of encoders (Tomczak & Welling, 2018), normalizing flows (Xu et al., 2019; Chen et al., 2016), resampled priors (Bauer & Mnih, 2019), and energy-based models (Pang et al., 2020; Vahdat et al., 2018b,a, 2020). Among them, energy-based models (EBMs) (Du & Mordatch, 2019; Pang et al., 2020) have shown promising results. However, they require running iterative MCMC during training which is computationally expensive when the energy function is represented by a neural network. Moreover, they scale poorly to hierarchical models where an EBM is defined on each group of latent variables.

Our key insight in this work is that a trainable prior is brought as close as possible to the aggregate posterior as a result of training a VAE. The mismatch between the prior and the aggregate posterior can be reduced by reweighting the prior to re-adjust its likelihood in the area of mismatch with the aggregate posterior. To represent this reweighting mechanism, we formulate the prior using an EBM that is defined by the product of a reweighting factor and a base trainable prior as shown in Fig. 6.1. We represent the reweighting factor using neural networks and the base prior using Normal distributions.

Instead of computationally expensive MCMC sampling, notorious for being slow, often sensitive to the choice of parameters (Du & Mordatch, 2019), we use noise contrastive estimation (NCE) (Gutmann & Hyvärinen, 2010) for training the EBM prior. We show that

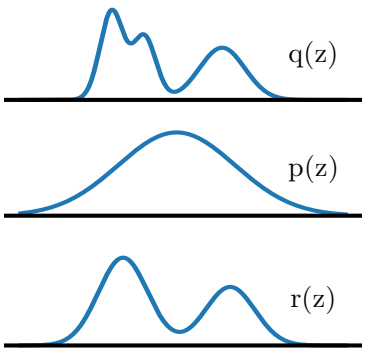



Figure 6.1: We propose an EBM prior using the product of a base prior $p(\mathbf{z})$ and a reweighting factor $r(\mathbf{z})$, designed to bring $p(\mathbf{z})$ closer to the aggregate posterior $q(\mathbf{z})$.

NCE trains the reweighting factor in our prior by learning a binary classifier to distinguish samples from a target distribution (i.e., approximate posterior) vs. samples from a noise distribution (i.e., the base trainable prior). However, since NCE's success depends on closeness of the noise distribution to the target distribution, we first train the VAE with the base prior to bring it close to the aggregate posterior. And then, we train the EBM prior using NCE.

In this work, we make the following contributions: i) We propose an EBM prior termed *noise contrastive prior (NCP)* which is trained by contrasting samples from the aggregate posterior to samples from a base prior. NCPs are simple and can be learned as a post-training mechanism to improve the expressivity of the prior. ii) We also show how NCPs are trained on hierarchical VAEs with many latent variable groups. We show that training hierarchical NCPs scales easily to many groups, as they are trained for each latent variable group in parallel. iii) Finally, we demonstrate that NCPs improve the generative quality of several forms of VAEs by a large margin across datasets.

## 6.2 BACKGROUND

We first review VAEs, their extension to hierarchical VAEs before discussing the prior hole problem.

**Variational Autoencoders:** VAEs learn a generative distribution $p(\mathbf{x}, \mathbf{z}) = p(\mathbf{z})p(\mathbf{x}|\mathbf{z})$ where $p(\mathbf{z})$ is a prior distribution over the latent variable $\mathbf{z}$ and $p(\mathbf{x}|\mathbf{z})$ is a likelihood function that generates the data $\mathbf{x}$ given $\mathbf{z}$. VAEs are trained by maximizing a variational lower bound $\mathcal{L}_{\text{VAE}}(\mathbf{x})$ on the log-likelihood $\log p(\mathbf{x}) \geq \mathcal{L}_{\text{VAE}}(\mathbf{x})$ where:

$$\mathcal{L}_{\text{VAE}}(\mathbf{x}) := \mathbb{E}_{q(\mathbf{z}|\mathbf{x})}[\log p(\mathbf{x}|\mathbf{z})] - \text{KL}(q(\mathbf{z}|\mathbf{x})||p(\mathbf{z})), \tag{6.1}$$

$q(\mathbf{z}|\mathbf{x})$ is an approximate posterior and KL is the Kullback–Leibler divergence. The final training objective is $\mathbb{E}_{p_d(\mathbf{x})}[\mathcal{L}_{\text{VAE}}(\mathbf{x})]$ where $p_d(\mathbf{x})$ is the data distribution (Kingma & Welling, 2014).

**Hierarchical VAEs (HVAEs):** To increase the expressivity of both prior and approximate posterior, earlier work adapted a hierarchical latent variable structure (Vahdat & Kautz, 2020; Child, 2021; Kingma et al., 2016; Sønderby et al., 2016; Gregor et al., 2016). In HVAEs, the latent variable $\mathbf{z}$ is divided into $K$ separate *groups*, $\mathbf{z} = \{\mathbf{z}_1, \ldots, \mathbf{z}_K\}$.

The approximate posterior and the prior distributions are then defined by $q(\mathbf{z}|\mathbf{x}) = \prod_{k=1}^{K} q(\mathbf{z}_k|\mathbf{z}_{<k}, \mathbf{x})$ and $p(\mathbf{z}) = \prod_{k=1}^{K} p(\mathbf{z}_k|\mathbf{z}_{<k})$. Using these, the training objective becomes:

$$\mathcal{L}_{\text{HVAE}}(\mathbf{x}) := \mathbb{E}_{q(\mathbf{z}|\mathbf{x})}[\log p(\mathbf{x}|\mathbf{z})] - \sum_{k=1}^{K} \mathbb{E}_{q(\mathbf{z}_{<k}|\mathbf{x})} \left[\text{KL}(q(\mathbf{z}_k|\mathbf{z}_{<k}, \mathbf{x})||p(\mathbf{z}_k|\mathbf{z}_{<k}))\right], \tag{6.2}$$

where $q(\mathbf{z}_{<k}|\mathbf{x}) = \prod_{i=1}^{k-1} q(\mathbf{z}_i|\mathbf{z}_{<i}, \mathbf{x})$ is the approximate posterior up to the $(k-1)^{\text{th}}$ group[1].

**Prior Hole Problem:** Let $q(\mathbf{z}) \triangleq \mathbb{E}_{p_d(\mathbf{x})}[q(\mathbf{z}|\mathbf{x})]$ denote the aggregate (approximate) posterior. In Sec. 6.4.1, we show that maximizing $\mathbb{E}_{p_d(\mathbf{x})}[\mathcal{L}_{\text{VAE}}(\mathbf{x})]$ w.r.t. the prior parameters corresponds to bringing the prior as close as possible to the aggregate posterior by minimizing $\text{KL}(q(\mathbf{z})||p(\mathbf{z}))$ w.r.t. $p(\mathbf{z})$. Formally, the prior hole problem refers to the phenomenon that $p(\mathbf{z})$ fails to match $q(\mathbf{z})$.

## 6.3 TRAINING ENERGY-BASED PRIORS USING MCMC

In this section, we show how a VAE with energy-based model in its prior can be trained. Assuming that the prior is in the form $p_{\text{EBM}}(\mathbf{z}) = \frac{1}{Z} r(\mathbf{z}) p(\mathbf{z})$, the variational bound is of the form:

$$\begin{aligned}
\mathbb{E}_{p_d(\mathbf{x})}[\mathcal{L}_{\text{VAE}}] &= \mathbb{E}_{p_d(\mathbf{x})} \left[\mathbb{E}_{q(\mathbf{z}|\mathbf{x})}[\log p(\mathbf{x}|\mathbf{z})] - \text{KL}(q(\mathbf{z}|\mathbf{x})||p_{\text{EBM}}(\mathbf{z}))\right] \\
&= E_{p_d(\mathbf{x})} \left[\mathbb{E}_{q(\mathbf{z}|\mathbf{x})}[\log p(\mathbf{x}|\mathbf{z}) - \log q(\mathbf{z}|\mathbf{x}) + \log r(\mathbf{z}) + \log p(\mathbf{z})]\right] - \log Z,
\end{aligned}$$

where the expectation term, similar to VAEs, can be trained using the reparameterization trick. The only problematic term is the log-normalization constant $\log Z$, which captures the gradient with respect to the parameters of the prior $p_{\text{EBM}}(\mathbf{z})$. Denoting these parameters by

[1]For $k = 1$, the expectation inside the summation is simplified to $\text{KL}(q(\mathbf{z}_1|\mathbf{x})||p(\mathbf{z}_1))$.

$\theta$, the gradient of $\log Z$ is obtained by:

$$\frac{\partial}{\partial \theta} \log Z = \frac{1}{Z} \int \frac{\partial (r(\mathbf{z})p(\mathbf{z}))}{\partial \theta} d\mathbf{z} = \int \frac{r(\mathbf{z})p(\mathbf{z})}{Z} \frac{\partial \log (r(\mathbf{z})p(\mathbf{z}))}{\partial \theta} d\mathbf{z} = \mathbb{E}_{P_{EBM}(\mathbf{z})}[\frac{\partial \log (r(\mathbf{z})p(\mathbf{z}))}{\partial \theta}], \tag{6.3}$$

where the expectation can be estimated using MCMC sampling from the EBM prior.

## 6.4 MAXIMIZING THE VARIATIONAL BOUND FROM THE PRIOR'S PERSPECTIVE

In this section, we discuss how maximizing the variational bound in VAEs from the prior's perspective corresponds to minimizing a KL divergence from the aggregate posterior to the prior. Note that this relation has been explored by Hoffman & Johnson (2016); Rezende & Viola (2018); Tomczak & Welling (2018) and we include it here for completeness.

### 6.4.1 VAE with a Single Group of Latent Variables

Denote the aggregate (approximate) posterior by $q(\mathbf{z}) \triangleq \mathbb{E}_{p_d(\mathbf{x})}[q(\mathbf{z}|\mathbf{x})]$. Here, we show that maximizing the $\mathbb{E}_{p_d(\mathbf{x})}[\mathcal{L}_{\text{VAE}}(\mathbf{x})]$ with respect to the prior parameters corresponds to learning the prior by minimizing $\text{KL}(q(\mathbf{z})||p(\mathbf{z}))$. To see this, note that the prior $p(\mathbf{z})$ only participates in the KL term in $\mathcal{L}_{\text{VAE}}$ (Eq. 6.1). We hence have:

$$\begin{aligned}
\arg\max_{p(\mathbf{z})} \mathbb{E}_{p_d(\mathbf{x})}[\mathcal{L}_{\text{VAE}}(\mathbf{x})] &= \arg\min_{p(\mathbf{z})} \mathbb{E}_{p_d(\mathbf{x})}[\text{KL}(q(\mathbf{z}|\mathbf{x})||p(\mathbf{z}))] \\
&= \arg\min_{p(\mathbf{z})} -\mathbb{E}_{p_d(\mathbf{x})}[H(q(\mathbf{z}|\mathbf{x}))] - \mathbb{E}_{q(\mathbf{z})}[\log p(\mathbf{z})] \\
&= \arg\min_{p(\mathbf{z})} -H(q(\mathbf{z})) - \mathbb{E}_{q(\mathbf{z})}[\log p(\mathbf{z})] \\
&= \arg\min_{p(\mathbf{z})} \text{KL}(q(\mathbf{z})||p(\mathbf{z})),
\end{aligned}$$

where $H(.)$ denotes the entropy. Above, we replaced the expected entropy $\mathbb{E}_{p_d(\mathbf{x})}[H(q(\mathbf{z}|\mathbf{x}))]$ with $H(q(\mathbf{z}))$ as the minimization is with respect to the parameters of the prior $p(\mathbf{z})$.

### 6.4.2 Hierarchical VAEs

Denote hierarchical approximate posterior and prior distributions by: $q(\mathbf{z}|\mathbf{x}) = \prod_{k=1}^{K} q(\mathbf{z}_k|\mathbf{z}_{<k}, \mathbf{x})$ and $p(\mathbf{z}) = \prod_{k=1}^{K} p(\mathbf{z}_k|\mathbf{z}_{<k})$. The hierarchical VAE objective becomes:

$$\mathcal{L}_{\text{HVAE}}(\mathbf{x}) = \mathbb{E}_{q(\mathbf{z}|\mathbf{x})}[\log p(\mathbf{x}|\mathbf{z})] - \sum_{k=1}^{K} \mathbb{E}_{q(\mathbf{z}_{<k}|\mathbf{x})} \left[\text{KL}(q(\mathbf{z}_k|\mathbf{z}_{<k}, \mathbf{x})||p(\mathbf{z}_k|\mathbf{z}_{<k}))\right], \tag{6.4}$$

where $q(\mathbf{z}_{<k}|\mathbf{x}) = \prod_{i=1}^{k-1} q(\mathbf{z}_i|\mathbf{z}_{<i}, \mathbf{x})$ is the approximate posterior up to the $(k-1)^{\text{th}}$ group. Denote the aggregate posterior up to the $(K-1)$

th*groupby*q(z˙¡k) $\triangleq \mathbb{E}_{p_d(\mathbf{x})}[q(\mathbf{z}_{<K}|\mathbf{x})]$ and the aggregate conditional for the $k$

th*groupgiventhepreviousgroups*q(z˙k—z˙¡k) $\triangleq \mathbb{E}_{p_d(\mathbf{x})}[q(\mathbf{z}_k|\mathbf{z}_{<k}, \mathbf{x})]$.

Here, we show that maximizing $\mathbb{E}_{p_d(\mathbf{x})}[\mathcal{L}_{\text{HVAE}}(\mathbf{x})]$ with respect to the prior corresponds to learning the prior by minimizing $\mathbb{E}_{q(\mathbf{z}_{<k})}\left[\text{KL}(q(\mathbf{z}_k|\mathbf{z}_{<k})||p(\mathbf{z}_k|\mathbf{z}_{<k}))\right]$ for each conditional:

$$\begin{aligned}
\arg\max_{p(\mathbf{z}_k|\mathbf{z}_{<k})} \mathbb{E}_{p_d(\mathbf{x})}[\mathcal{L}_{\text{HVAE}}(\mathbf{x})] &= \arg\min_{p(\mathbf{z}_k|\mathbf{z}_{<k})} \mathbb{E}_{p_d(\mathbf{x})} \left[\mathbb{E}_{q(\mathbf{z}_{<k}|\mathbf{x})} \left[\text{KL}(q(\mathbf{z}_k|\mathbf{z}_{<k}, \mathbf{x})||p(\mathbf{z}_k|\mathbf{z}_{<k}))\right]\right] \\
&= \arg\min_{p(\mathbf{z}_k|\mathbf{z}_{<k})} -\mathbb{E}_{p_d(\mathbf{x})q(\mathbf{z}_{<k}|\mathbf{x})q(\mathbf{z}_k|\mathbf{z}_{<k},\mathbf{x})} \left[\log p(\mathbf{z}_k|\mathbf{z}_{<k})\right] \\
&= \arg\min_{p(\mathbf{z}_k|\mathbf{z}_{<k})} -\mathbb{E}_{q(\mathbf{z}_k,\mathbf{z}_{<k})} \left[\log p(\mathbf{z}_k|\mathbf{z}_{<k})\right] \\
&= \arg\min_{p(\mathbf{z}_k|\mathbf{z}_{<k})} -\mathbb{E}_{q(\mathbf{z}_{<k})} \left[\mathbb{E}_{q(\mathbf{z}_k|\mathbf{z}_{<k})} \left[\log p(\mathbf{z}_k|\mathbf{z}_{<k})\right]\right] \\
&= \arg\min_{p(\mathbf{z}_k|\mathbf{z}_{<k})} \mathbb{E}_{q(\mathbf{z}_{<k})} \left[-H(q(\mathbf{z}_k|\mathbf{z}_{<k})) - \mathbb{E}_{q(\mathbf{z}_k|\mathbf{z}_{<k})} \left[\log p(\mathbf{z}_k|\mathbf{z}_{<k})\right]\right] \\
&= \arg\min_{p(\mathbf{z}_k|\mathbf{z}_{<k})} \mathbb{E}_{q(\mathbf{z}_{<k})} \left[\text{KL}(q(\mathbf{z}_k|\mathbf{z}_{<k})||p(\mathbf{z}_k|\mathbf{z}_{<k}))\right].
\end{aligned} \tag{6.5}$$

## 6.5 NOISE CONTRASTIVE PRIORS (NCPS)

One of the main causes of the prior hole problem is the limited expressivity of prior that prevents it from matching the aggregate posterior. Recently, EBMs have shown promising results in representing complex distributions. Motivated by their success, we introduce the noise contrastive prior (NCP)

$p_{\text{NCP}}(\mathbf{z}) = \frac{1}{Z} r(\mathbf{z}) p(\mathbf{z})$, where $p(\mathbf{z})$ is a base prior distribution, e.g., a Normal, $r(\mathbf{z})$ is a reweighting factor, and $Z = \int r(\mathbf{z}) p(\mathbf{z}) d\mathbf{z}$ is the normalization constant. The function $r : \mathbb{R}^n \to \mathbb{R}^+$ maps the latent variable $\mathbf{z} \in \mathbb{R}^n$ to a positive scalar, and can be implemented using neural nets.

The reweighting factor $r(\mathbf{z})$ can be trained using MCMC as discussed in Sec. 6.3. However,

MCMC requires expensive sampling iterations that scale poorly to hierarchical VAEs. To address this, we describe a noise contrastive estimation based approach to train $p_{\text{NCP}}(\mathbf{z})$ without MCMC.

### 6.5.1 Conditional NCE for Hierarchical VAEs

In this section, we describe how we derive the NCE training objective for hierarchical VAEs given in Eq. (6.11). Our goal is to learn the likelihood ratio between the aggregate conditional $q(\mathbf{z}_k|\mathbf{z}_{<k})$ and the prior $p(\mathbf{z}_k|\mathbf{z}_{<k})$. We can define the NCE objective to train the discriminator $D_k(\mathbf{z}_k, \mathbf{z}_{<k})$ that classifies $\mathbf{z}_k$ given samples from the previous groups $\mathbf{z}_{<k}$ using:

$$\min_{D_k} \; - \mathbb{E}_{q(\mathbf{z}_k|\mathbf{z}_{<k})}[\log D_k(\mathbf{z}_k, \mathbf{z}_{<k})] - \mathbb{E}_{p(\mathbf{z}_k|\mathbf{z}_{<k})}[\log(1 - D_k(\mathbf{z}_k, \mathbf{z}_{<k}))] \quad \forall \mathbf{z}_{<k}. \tag{6.6}$$

Since $\mathbf{z}_{<k}$ is in a high dimensional space, we cannot apply the minimization $\forall \mathbf{z}_{<k}$. Instead, we sample from $\mathbf{z}_{<k}$ using the aggregate approximate posterior $q(\mathbf{z}_{<k})$ as done for the KL in a hierarchical model (Eq. (6.5)):

$$\min_{D_k} \; \mathbb{E}_{q(\mathbf{z}_{<k})}\Big[ - \; \mathbb{E}_{q(\mathbf{z}_k|\mathbf{z}_{<k})}[\log D_k(\mathbf{z}_k, \mathbf{z}_{<k})] - \mathbb{E}_{p(\mathbf{z}_k|\mathbf{z}_{<k})}[\log(1 - D_k(\mathbf{z}_k, \mathbf{z}_{<k}))]\Big]. \tag{6.7}$$

Since $q(\mathbf{z}_{<k})q(\mathbf{z}_k|\mathbf{z}_{<k}) = q(\mathbf{z}_k, \mathbf{z}_{<k}) = \mathbb{E}_{p_d(\mathbf{x})}[q(\mathbf{z}_{<k}|\mathbf{x})q(\mathbf{z}_k|\mathbf{z}_{<k}, \mathbf{x})]$, we have:

$$\min_{D_k} \; \mathbb{E}_{p_d(\mathbf{x})q(\mathbf{z}_{<k}|\mathbf{x})}\Big[ - \; \mathbb{E}_{q(\mathbf{z}_k|\mathbf{z}_{<k}, \mathbf{x})}[\log D_k(\mathbf{z}_k, \mathbf{z}_{<k})] - \mathbb{E}_{p(\mathbf{z}_k|\mathbf{z}_{<k})}[\log(1 - D_k(\mathbf{z}_k, \mathbf{z}_{<k}))]\Big]. \tag{6.8}$$

Finally, instead of passing all the samples from the previous latent variables groups to $D$, we can pass the context feature $c(\mathbf{z}_{<k})$ that extracts a representation from all the previous groups:

$$\min_{D_k} \; \mathbb{E}_{p_d(\mathbf{x})q(\mathbf{z}_{<k}|\mathbf{x})}\Big[ - \; \mathbb{E}_{q(\mathbf{z}_k|\mathbf{z}_{<k}, \mathbf{x})}[\log D_k(\mathbf{z}_k, c(\mathbf{z}_{<k}))] - \mathbb{E}_{p(\mathbf{z}_k|\mathbf{z}_{<k})}[\log(1 - D_k(\mathbf{z}_k, c(\mathbf{z}_{<k})))]\Big]. \tag{6.9}$$

### 6.5.2 Learning The Reweighting Factor with Noise Contrastive Estimation

Recall that training VAEs closes the gap between the prior and the aggregate posterior by minimizing $\text{KL}(q(\mathbf{z})||p(\mathbf{z}))$ with respect to prior. Assuming the base prior $p(\mathbf{z})$ to be fixed, $\text{KL}(q(\mathbf{z})||p_{\text{NCP}}(\mathbf{z}))$ is zero when $r(\mathbf{z}) = q(\mathbf{z})/p(\mathbf{z})$. However, since we do not have the density function for $q(\mathbf{z})$, we cannot compute the ratio explicitly. Instead, in this work, we propose to

estimate $r(\mathbf{z})$ using noise contrastive estimation (Gutmann & Hyvärinen, 2010), also known as the likelihood ratio trick that has been popularized in machine learning by predictive coding (Oord et al., 2018) and generative adversarial networks (GANs) (Goodfellow et al., 2014b). Since, we can generate samples from both $p(\mathbf{z})$ and $q(\mathbf{z})$[2], we train a binary classifier to distinguish samples from $q(\mathbf{z})$ and samples from the base prior $p(\mathbf{z})$ by minimizing the binary cross-entropy loss:

$$\min_{D} - \mathbb{E}_{\mathbf{z}\sim q(\mathbf{z})}[\log D(\mathbf{z})] - \mathbb{E}_{\mathbf{z}\sim p(\mathbf{z})}[\log(1 - D(\mathbf{z}))]. \tag{6.10}$$

Here, $D : \mathbb{R}^n \to (0, 1)$ is a binary classifier that generates the classification prediction probabilities. Eq. (6.10) is minimized when $D(\mathbf{z}) = \frac{q(\mathbf{z})}{q(\mathbf{z})+p(\mathbf{z})}$. Denoting the classifier at optimality by $D^*(\mathbf{z})$, we estimate the reweighting factor $r(\mathbf{z}) = \frac{q(\mathbf{z})}{p(\mathbf{z})} \approx \frac{D^*(\mathbf{z})}{1-D^*(\mathbf{z})}$. The appealing advantage of this estimator is that it is obtained by simply training a binary classifier rather than using expensive MCMC sampling.

### 6.5.3 Two-stage Training for Noise Contrastive Priors

To properly learn the reweighting factor, NCE training requires the base prior distribution to be close to the target distribution. Intuitively, if $p(\mathbf{z})$ is very close to $q(\mathbf{z})$ (i.e., $p(\mathbf{z}) \approx q(\mathbf{z})$), the optimal classifier will have a large loss value in Eq. (6.10), and we will have $r(\mathbf{z}) \approx 1$. If $p(\mathbf{z})$ is instead far from $q(\mathbf{z})$, the binary classifier will easily learn to distinguish samples from the two distributions and it will not learn the likelihood ratios correctly. If $p(\mathbf{z})$ is roughly close to $q(\mathbf{z})$, then the binary classifier can learn the ratios.

To ensure that the base prior distribution is close to the target aggregate posterior, we propose a two-stage training algorithm. In the first stage, we train the VAE with only the base prior $p(\mathbf{z})$. From Sec. 6.4.1, we know that at the end of training, $p(\mathbf{z})$ is as close as possible to $q(\mathbf{z})$. In the second stage, we freeze the VAE model including the approximate posterior $q(\mathbf{z}|\mathbf{x})$, the base prior $p(\mathbf{z})$, and the likelihood $p(\mathbf{x}|\mathbf{z})$, and we only train the reweighting factor $r(\mathbf{z})$ using Eq. (6.10). The second stage can be thought of as replacing the base distribution $p(\mathbf{z})$ with a more expressive distribution of the form $p_{\text{NCP}}(\mathbf{z}) \propto r(\mathbf{z})p(\mathbf{z})$. Hence, NCP matches the prior to the aggregate posterior $q(\mathbf{z})$ using $r(\mathbf{z})$. Note that our proposed method is generic as it only assumes that we can draw samples from $q(\mathbf{z})$ and $p(\mathbf{z})$, which applies to any VAE. Our training is illustrated in Fig. 6.2.

[2]We generate samples from the aggregate posterior $q(\mathbf{z}) = \mathbb{E}_{p_d(\mathbf{x})}[q(\mathbf{z}|\mathbf{x})]$ via ancestral sampling: draw data from the training set ($\mathbf{x} \sim p_d(\mathbf{x})$) and then sample from $\mathbf{z} \sim q(\mathbf{z}|\mathbf{x})$.

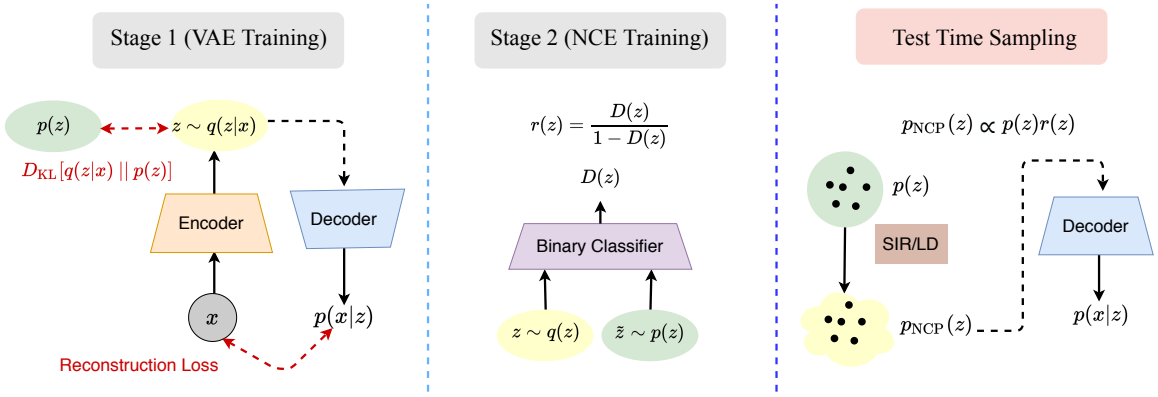



Figure 6.2: NCP-VAE is trained in two stages. In the first stage, we train a VAE using the original VAE objective. In the second stage, we train the reweighting factor $r(\mathbf{z})$ using noise contrastive estimation (NCE). NCE trains a classifier to distinguish samples from the prior and samples from the aggregate posterior. Our noise contrastive prior (NCP) is then constructed by the product of the base prior and the reweighting factor, formed via the classifier. At test time, we sample from NCP using SIR or LD. These samples are then passed to the decoder to generate output samples.

### 6.5.4 Test Time Sampling

To sample from a VAE with an NCP, we first generate samples from the NCP and pass them to the decoder to generate output samples (Fig. 6.2). We propose two methods for sampling from NCPs.

**Sampling-Importance-Resampling (SIR):** We first generate $M$ samples from the base prior distribution $\{\mathbf{z}^{(m)}\}_{m=1}^{M} \sim p(\mathbf{z})$. We then resample one of the $M$ proposed samples using importance weights proportional to $w^{(m)} = p_{\mathrm{NCP}}(\mathbf{z}^{(m)})/p(\mathbf{z}^{(m)}) = r(\mathbf{z}^{(m)})$. The benefit of this technique: both proposal generation and the evaluation of $r$ on the samples are done in parallel.

**Langevin Dynamics (LD):** Since our NCP is an EBM, we can use LD for sampling. Denoting the energy function by $E(\mathbf{z}) = -\log r(\mathbf{z}) - \log p(\mathbf{z})$, we initialize a sample $\mathbf{z}_0$ by drawing from $p(\mathbf{z})$ and update the sample iteratively using: $\mathbf{z}_{t+1} = \mathbf{z}_t - 0.5\ \lambda \nabla_{\mathbf{z}} E(\mathbf{z}) + \sqrt{\lambda}\epsilon_t$ where $\epsilon_t \sim \mathcal{N}(0, 1)$ and $\lambda$ is the step size. LD is run for a finite number of iterations, and in contrast to SIR, it is slower given its sequential form.

### 6.5.5 Generalization to Hierarchical VAEs

The state-of-the-art VAEs (Child, 2021; Vahdat & Kautz, 2020) use a hierarchical $q(\mathbf{z}|\mathbf{x})$ and $p(\mathbf{z})$. Here $p(\mathbf{z})$ is chosen to be a Gaussian distribution. Sec. 6.4.2 shows that training a HVAE encourages the prior to minimize $\mathbb{E}_{q(\mathbf{z}_{<k})}\left[\mathrm{KL}(q(\mathbf{z}_k|\mathbf{z}_{<k})||p(\mathbf{z}_k|\mathbf{z}_{<k}))\right]$ for each

conditional, where $q(\mathbf{z}_{<k}) \triangleq \mathbb{E}_{p_d(\mathbf{x})}[q(\mathbf{z}_{<K}|\mathbf{x})]$ is the aggregate posterior up to the $(k-1)$ th*group*, *and*q(z˙k—z˙¡k) $\triangleq$ $\mathbb{E}_{p_d(\mathbf{x})}[q(\mathbf{z}_k|\mathbf{z}_{<k}, \mathbf{x})]$ is the aggregate conditional for the $k^{\text{th}}$ group.

Given this observation, we extend NCPs to hierarchical models to match each conditional in the prior with $q(\mathbf{z}_k|\mathbf{z}_{<k})$. Formally, we define hierarchical NCPs by $p_{\text{NCP}}(\mathbf{z}) = \frac{1}{Z}\prod_{k=1}^{K} r(\mathbf{z}_k|\mathbf{z}_{<k})p(\mathbf{z}_k|\mathbf{z}_{<k})$ where each factor is an EBM. $p_{\text{NCP}}(\mathbf{z})$ resembles EBMs with autoregressive structure among groups (Nash & Durkan, 2019).

In the first stage, we train the HVAE with prior $\prod_{k=1}^{K} p(\mathbf{z}_k|\mathbf{z}_{<k})$. For the second stage, we use $K$ binary classifiers, each for a hierarchical group. Following Sec. 6.5.1, we train each classifier via:

$$\min_{D_k} \mathbb{E}_{p_d(\mathbf{x})q(\mathbf{z}_{<k}|\mathbf{x})}\Big[ - \mathbb{E}_{q(\mathbf{z}_k|\mathbf{z}_{<k},\mathbf{x})}[\log D_k(\mathbf{z}_k, c(\mathbf{z}_{<k}))] - \mathbb{E}_{p(\mathbf{z}_k|\mathbf{z}_{<k})}[\log(1 - D_k(\mathbf{z}_k, c(\mathbf{z}_{<k})))]\Big], \tag{6.11}$$

where the outer expectation samples from groups up to the $(k-1)^{\text{th}}$ group, and the inner expectations sample from approximate posterior and base prior for the $k^{\text{th}}$ group, conditioned on the same $\mathbf{z}_{<k}$. The discriminator $D_k$ classifies samples $\mathbf{z}_k$ while conditioning its prediction on $\mathbf{z}_{<k}$ using a shared context feature $c(\mathbf{z}_{<k})$.

The NCE training in Eq. (6.11) is minimized when $D_k(\mathbf{z}_k, c(\mathbf{z}_{<k})) = \frac{q(\mathbf{z}_k|\mathbf{z}_{<k})}{q(\mathbf{z}_k|\mathbf{z}_{<k})+p(\mathbf{z}_k|\mathbf{z}_{<k})}$. Denoting the classifier at optimality by $D_k^*(\mathbf{z}, c(\mathbf{z}_{<k}))$, we obtain the reweighting factor $r(\mathbf{z}_k|\mathbf{z}_{<k}) \approx \frac{D_k^*(\mathbf{z}_k, c(\mathbf{z}_{<k}))}{1-D_k^*(\mathbf{z}_k, c(\mathbf{z}_{<k}))}$ in the second stage. Given our hierarchical NCP, we use ancestral sampling to sample from the prior. For sampling from each group, we can use SIR or LD as discussed before.

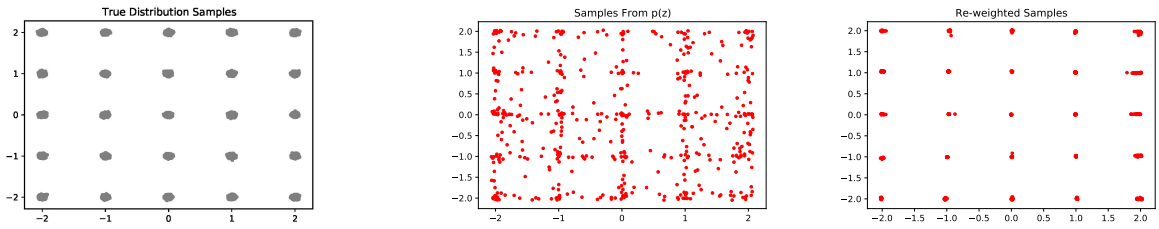
The context feature $c(\mathbf{z}_{<k})$ extracts a representation from $\mathbf{z}_{<k}$. Instead of learning a new representation at stage two, we simply use the representation that is extracted from $\mathbf{z}_{<k}$ in the hierarchical prior, trained in the first stage.

Note that the binary classifiers are trained in parallel for all groups.

## 6.6 EXPERIMENTS

In this section, we first examine the efficacy of our approach on a 2D toy dataset in Sec. 6.6.1. We then situate NCP against prior art on several commonly used single group VAE models in Sec. 6.6.2. Finally, in Sec. 6.6.3, we present our main results where we apply NCP to hierarchical NVAE (Vahdat & Kautz, 2020) to demonstrate that our approach can be applied to large scale models successfully.

In most our experiments, we measure the sample quality using the Fréchet Inception Distance (FID) score (Heusel et al., 2017) with 50,000 samples, as computing the log-likelihood




(a) Samples from the true distribution (b) Samples from VAE (c) Samples from NCP-VAE

Figure 6.3: Qualitative results on mixture of 25-Gaussians. Fig.(a) shows the true distribution samples. Note that the samples decoded from the base prior $p(\mathbf{z})$ without the NCP approach generates many points in the the low density regions under the data distribution, (Fig. (b)). These points are removed using our NCP approach (Fig. (c)).

requires estimating the intractable normalization constant. For generating samples from the model, we use SIR with 5K proposal samples. To report log-likelihood results, we train models with small latent space only on the dynamically binarized MNIST (LeCun, 1998) dataset. We intentionally limit the latent space to ensure that we can estimate the normalization constant correctly.

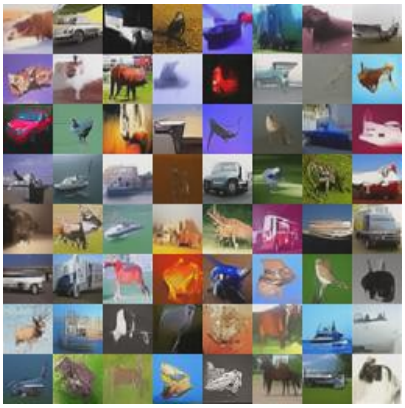
### 6.6.1 Experiment on Synthetic Data

In Fig. 6.3, we demonstrate the efficacy of our approach on the 25-Gaussians dataset, that is generated by a mixture of 25 two-dimensional Gaussian distributions, arranged on a grid. The encoder and decoder of the VAE have 4 fully connected layers with 256 hidden units, with 20 dimensional latent variables. The discriminator has 4 fully connected layers with 256 hidden units. Note that many samples decoded from the base prior $p(\mathbf{z})$ (Fig. 6.3(b)) are located at the low density regions in the data distribution. These samples are removed using our NCP approach (Fig. 6.3(c)). We use 50k samples from the true distribution to estimate the log-likelihood. Our NCP-VAE obtains an average test log-likelihood of $-0.954$ nats compared to $-2.753$ nats obtained by vanilla VAE.

### 6.6.2 Comparison to Prior Art

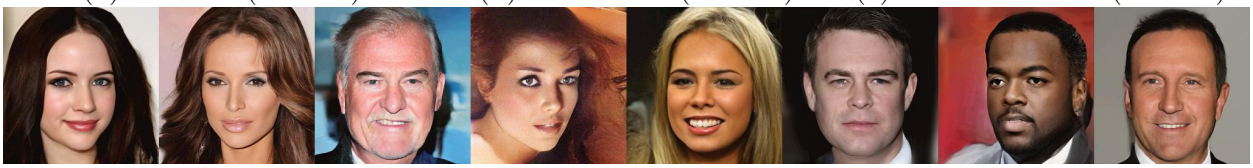
In this section, we apply NCP to several commonly used small VAE models. Our goal, here, is to situate our proposed model against (i) two-stage VAE models that train a (variational) autoencoder first, and then, fit a prior distribution (Sec. 6.6.2), and (ii) VAEs with reweighted priors (Sec. 6.6.2). To make sure that these comparisons are fair, we follow exact training setup and network architectures from the previous work as discussed below.

(a) MNIST ($t = 1.0$)



(b) CIFAR-10 ($t = 0.5$)

(c) CelebA 64×64 ($t = 0.7$)



(d) CelebA HQ 256×256 ($t = 0.7$)

Figure 6.4: Randomly sampled images from NCP-VAE with the temperature $t$ for the prior.

| Model | FID↓ |
|---|---|
| VAE w/ Gaussian prior | 48.12† |
| 2s-VAE (Dai & Wipf, 2018) | 49.70† |
| WAE (Tolstikhin et al., 2018) | 42.73† |
| RAE (Ghosh et al., 2020) | 40.95† |
| NCP w/ Gaussian prior as base | 41.28 |
| NCP w/ GMM prior as base | **39.00** |
| Base VAE-Recon | 36.01 |

Table 6.1: Comparison with two-stage VAEs on CelebA-64 with RAE (Ghosh et al., 2020) networks. † Results reported by Ghosh et al. (2020).

| Model | NLL↓ |
|---|---|
| VAE w/ Gaussian prior | 84.82 |
| VAE w/ LARS prior (Bauer & Mnih, 2019) | 83.03 |
| VAE w/ SNIS prior (Lawson et al., 2019) | **82.52** |
| NCP-VAE | 82.82 |

Table 6.2: Likelihood results on MNIST on single latent group model with architecture from LARS (Bauer & Mnih, 2019) & SNIS (Lawson et al., 2019) (results in nats). We closely follow the training hyperparameters used by Lawson et al. (2019)

Comparison against Two-Stage VAEs

Here, we show the generative performance of our approach applied to the VAE architecture in RAE (Ghosh et al., 2020) on the CelebA-64 dataset (Liu et al., 2015). We borrow the exact training setup from Ghosh et al. (2020) and implement our method using their publicly available code[3]. Note that this VAE architecture has only one latent variable group. The same base architecture was used in the implementation of 2s-VAE (Dai & Wipf, 2018) and WAE (Tolstikhin et al., 2018). In order to compare our method to these models, we use the reported results from RAE (Ghosh et al., 2020). We apply our NCP-VAE on top of both vanilla VAE with a Gaussian prior and a 10-component Gaussian mixture model (GMM) prior that was proposed in RAEs. As we can see in Tab. 6.1, our NCP-VAE improves the performance of the base VAE, improving the FID score to 41.28 from 48.12. Additionally, when NCP is applied to the VAE with GMM prior (the RAE model), it improves its performance from 40.95 to the FID score of 39.00. We also report the FID score for reconstructed images using samples from the aggregate posterior $q(\mathbf{z})$ instead of the prior. Note that this value represents the best FID score that one can obtain by perfectly matching the prior to the the aggregate posterior in the second stage. The high FID score of 36.01 indicates that the small VAEs cannot reconstruct data samples well due to the small network architecture and latent space. Thus, even with expressive priors FID for two-stage VAEs are lower bounded by 36.01 in the $2^{nd}$ stage.

Comparison against Reweighted Priors

LARS (Bauer & Mnih, 2019) and SNIS (Lawson et al., 2019) train reweighted priors similar to our EBM prior. To compare NCP-VAE against these methods, we implement our method using the VAE and energy-function networks from Lawson et al. (2019). We closely follow the training hyperparameters used by Lawson et al. (2019) as well as their approach for obtaining a lower bound on the log likelihood. As shown in Tab. 6.2, NCP-VAE obtains the negative log-likelihood (NLL) of 82.82, comparable to Lawson et al. (2019), while outperforming LARS (Bauer & Mnih, 2019). Although NCP-VAE is slightly inferior to SNIS on MNIST, it has several advantages as discussed in Sec. 6.7.

[3] https://github.com/ParthaEth/Regularized_autoencoders-RAE-

Training using Normalizing Flows

Chen et al. (2016) (Sec. 3.2) show that a normalizing flow in the approximate posterior is equivalent to having its inverse in the prior. The base NVAE uses normalizing flows in the encoder. As a part of VAE training, prior and aggregate posterior are brought close, i.e., normalizing flows are implicitly used. We argue that normalizing flows provide limited gains to address the prior-hole problem (see Fig. 1 by Kingma et al. (2016)). Yet, our model further improves the base VAE equipped with normalizing flow.

| Model | FID↓ |
|---|---|
| NCP-VAE (ours) | **5.25** |
| VAEBM (Xiao et al., 2021) | 5.31 |
| NVAE (Vahdat & Kautz, 2020) | 13.48 |
| RAE (Ghosh et al., 2020) | 40.95 |
| 2s-VAE (Dai & Wipf, 2018) | 44.4 |
| WAE (Tolstikhin et al., 2018) | 35 |
| Perceptial AE(Zhang et al., 2020) | 13.8 |
| Latent EBM (Pang et al., 2020) | 37.87 |
| COCO-GAN (Lin et al., 2019) | 4.0 |
| QA-GAN (Parimala & Channappayya, 2019) | 6.42 |
| NVAE-Recon (Vahdat & Kautz, 2020) | 1.03 |

Table 6.3: Generative performance on CelebA-64

| Model | FID↓ |
|---|---|
| NCP-VAE (ours) | 24.08 |
| VAEBM (Xiao et al., 2021) | **12.96** |
| NVAE (Vahdat & Kautz, 2020) | 51.71 |
| RAE (Ghosh et al., 2020) | 74.16 |
| 2s-VAE (Dai & Wipf, 2018) | 72.9 |
| Perceptial AE (Zhang et al., 2020) | 51.51 |
| EBM (Du & Mordatch, 2019) | 40.58 |
| Latent EBM (Pang et al., 2020) | 70.15 |
| Style-GANv2 (Karras et al., 2020) | 3.26 |
| DDPM (Ho et al., 2020) | 3.17 |
| Score SDE Song et al. (2021) | 3.20 |
| NVAE-Recon (Vahdat & Kautz, 2020) | 2.67 |

Table 6.4: Generative performance on CIFAR-10

| Model | FID↓ |
|---|---|
| NCP-VAE (ours) | 24.79 |
| VAEBM (Xiao et al., 2021) | **20.38** |
| NVAE (Vahdat & Kautz, 2020) | 40.26 |
| GLOW (Kingma & Dhariwal, 2018) | 68.93 |
| Advers. LAE (Pidhorskyi et al., 2020) | 19.21 |
| PGGAN (Karras et al., 2017) | 8.03 |
| NVAE-Recon (Vahdat & Kautz, 2020) | 0.45 |

Table 6.5: Generative results on CelebA-HQ-256

| Model | NLL↓ |
|---|---|
| NCP-VAE (ours) | **78.10** |
| NVAE-small (Vahdat & Kautz, 2020) | 78.67 |
| BIVA (Maaløe et al., 2019) | 78.41 |
| DAVE++ (Vahdat et al., 2018b) | 78.49 |
| IAF-VAE (Kingma et al., 2016) | 79.10 |
| VampPrior AR dec. (Tomczak & Welling (2018)) | 78.45 |
| DVAE (Rolfe, 2016) | 80.15 |

Table 6.6: Likelihood results on MNIST in nats

### 6.6.3 Quantitative Results on Hierarchical Models

In this section, we apply NCP to the hierarchical VAE model proposed in NVAE (Vahdat & Kautz, 2020). We examine NCP-VAE on four datasets including dynamically binarized MNIST (LeCun, 1998), CIFAR-10 (Krizhevsky et al., 2009), CelebA-64 (Liu et al., 2015) and CelebA-HQ-256 (Karras et al., 2017). For CIFAR-10 and CelebA-64, the model has 30 groups, and for CelebA-HQ-256 it has 20 groups. For MNIST, we train an NVAE model with a small latent space on MNIST with 10 groups of $4\times4$ latent variables. The small latent space allows us to estimate the partition function confidently (std. of $\log Z$ estimation $\leq 0.23$). The quantitative results are reported in Table 6.3, Table 6.4, Table 6.5, and Table 6.6. On all four datasets, our model improves upon NVAE, and it reduces the gap with GANs by a large margin. On CelebA 64, we improve NVAE from an FID of 13.48 to 5.25, comparable to GANs. On CIFAR-10, NCP-VAE improves the NVAE FID of 51.71 to 24.08. On MNIST, although our latent space is much smaller, our model outperforms previous VAEs. NVAE has reported 78.01 nats on this dataset with a larger latent space.

On CIFAR-10 and CelebA-HQ-256, recently proposed VAEBM (Xiao et al., 2021) outperforms our NCP-VAE. However, we should note that (i) NCP-VAE and VAEBM are

| # groups | NVAE | NCP-VAE |
|---|---|---|
| 6 | 33.18 | 18.68 |
| 15 | 14.96 | 5.96 |
| 30 | 13.48 | **5.25** |

Table 6.7: Number of groups & generative performance in FID↓

complementary to each other, as NCP-VAE targets the latent space while VAEBM forms an EBM on the data space. We expect improvements by combining these two models. (ii) VAEBM assumes that the data lies on a continuous space whereas NCP-VAE does not make any such assumption and it can be applied to discrete data (like binarized MNIST in Table 6.6), graphs, and text. (iii) NCP-VAE is much simpler to setup as it involves training a binary classifier whereas VAEBM requires MCMC for both training and test.

**Reconstruction:**

Appendix A.4 shows nearest neighbours from training data for generated samples.

### 6.6.4 Qualitative Results

We visualize samples generated by NCP-VAE with the NVAE backbone in Fig. 6.4 without any manual intervention. We adopt the common practice of reducing the temperature of the base prior $p(\mathbf{z})$ by scaling down the standard-deviation of the conditional Normal distributions (Kingma & Dhariwal, 2018)[4].

Brock et al. (2018); Vahdat & Kautz (2020) also observe that re-adjusting the batch-normalization (BN), given a temperature applied to the prior, improves the generative quality. Similarly, we achieve diverse, high-quality images by re-adjusting the BN statistics as described by Vahdat & Kautz (2020).

Additional qualitative results are shown in Appendix A.6.

### 6.6.5 Additional Ablation Studies

We perform additional experiments to study i) how hierarchical NCPs perform as the number of latent groups increases, ii) the impact of SIR and LD hyperparameters, and iii) what the classification loss in NCE training conveys about $p(\mathbf{z})$ and $q(\mathbf{z})$. All experiments are performed on CelebA-64.

[4]Lowering the temperature is only used to obtain qualitative samples, not for the quantitative results in Sec. 6.6.

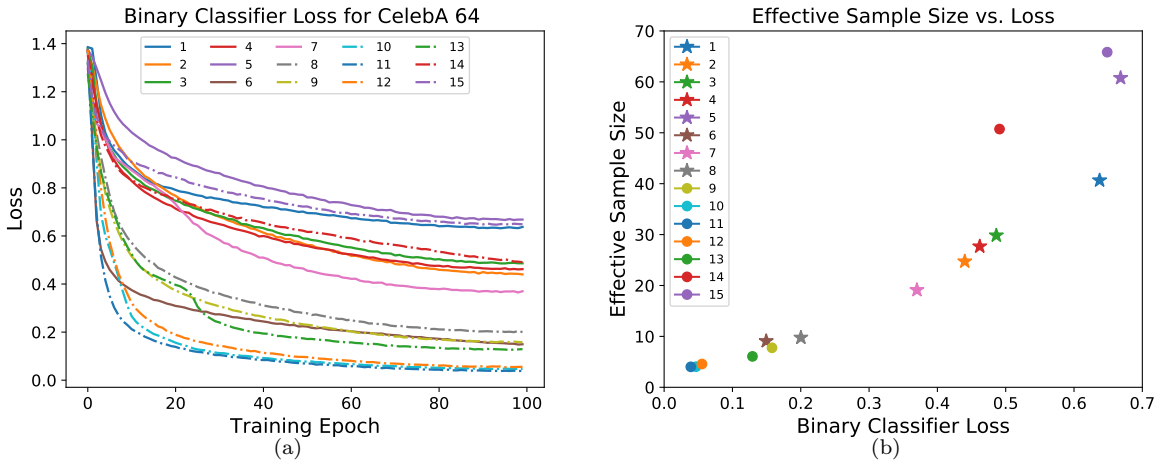



Figure 6.5: **(a)** Classification loss for binary classifiers on latent variable groups. A larger final loss upon training indicates that $q(\mathbf{z})$ and $p(\mathbf{z})$ are more similar. **(b)** The effective sample size vs. the final loss value at the end of training. Higher effective sample size implies similarity of two distributions.

**Number of latent variable groups:** Table 6.7 shows the generative performance of hierarchical NCP with different amounts of latent variable groups. As we increase the number of groups, the FID score of both NVAE and our model improves. This shows the efficacy of our NCPs, with expressive hierarchical priors in the presence of many groups.

**SIR and LD parameters:** The computationally complexity of SIR is similar to LD if we set the number of proposal samples in SIR equal to the number LD iterations. In Table 6.8 we observe that increasing both the number of proposal samples in SIR and the LD iterations leads to a noticeable improvement in FID score. For SIR, the proposal, generation and the evaluation of $r(\mathbf{z})$ are parallelizable. Hence, as shown in Table 6.8, image generation is faster with SIR than with LD (LD iterations are sequential). However, GPU memory usage scales with the number of SIR proposals, but not with the number of LD iterations. Interestingly, SIR, albeit simple, performs better than LD when using about the same compute.

**Classification loss in NCE:** We can draw a direct connection between the classification loss in Eq. 6.10 and the similarity of $p(\mathbf{z})$ and $q(\mathbf{z})$. Denoting the classification loss in Eq. 6.10 at optimality by $\mathcal{L}^*$, Goodfellow et al. (2014b) show that $\text{JSD}(p(\mathbf{z})||q(\mathbf{z})) = \log 2 - 0.5 \times \mathcal{L}^*$ where JSD denotes the Jensen–Shannon divergence between two distributions.

Fig. 6.5(a) plots the classification loss (Eq. 6.11) for each classifier for a 15-group NCP trained on the CelebA-64 dataset. Assume that the classifier loss at the end of training is a good approximation of $\mathcal{L}^*$. We observe that 8 out of 15 groups have $\mathcal{L}^* \geq 0.4$, indicating a good overlap between $p(\mathbf{z})$ and $q(\mathbf{z})$ for those groups.

To further assess the impact of the distribution match on SIR sampling, in Fig. 6.5(b),

we visualize the effective sample size (ESS)[5] in SIR vs. $\mathcal{L}^*$ for the same group. We observe a strong correlation between $\mathcal{L}^*$ and the effective sample size. SIR is more reliable on the same 8 groups that have high classification loss. These groups are at the top of the NVAE hierarchy which have been shown to control the global structure of generated samples (see B.6 in Vahdat & Kautz (2020)).

| # SIR proposal samples | FID↓ | Time-1 (sec) | Time-10 (sec) | Memory (GB) | # LD iterations | FID↓ | Time-1 (sec) | Time-10 (sec) | Memory (GB) |
|---|---|---|---|---|---|---|---|---|---|
| 5 | 11.75 | 0.34 | 0.42 | 1.96 | 5 | 14.44 | 3.08 | 3.07 | 1.94 |
| 50 | 8.58 | 0.40 | 1.21 | 4.30 | 50 | 12.76 | 27.85 | 28.55 | 1.94 |
| 500 | 6.76 | 1.25 | 9.43 | 20.53 | 500 | 8.12 | 276.13 | 260.35 | 1.94 |
| 5000 | **5.25** | 10.11 | 95.67 | 23.43 | 1000 | **6.98** | 552 | 561.44 | 1.94 |

Table 6.8: Effect of SIR sample size and LD iterations. Time-$N$ is the time used to generate a batch of $N$ images.

| # group | $(q,\, p)$ | $(q,\, p_{\text{NCP}})$ |
|---|---|---|
| 5 | 0.002 | 0.002 |
| 10 | 0.08 | 0.06 |
| 12 | 0.08 | 0.07 |

Table 6.9: MMD comparison

**Analysis of the re-weighting technique:** To show that samples from NCP ($p_{\text{NCP}}(\mathbf{z})$) are closer to the aggregate posterior $q(\mathbf{z})$ compared to the samples from the base prior $p(\mathbf{z})$, we take 5k samples from $q(\mathbf{z})$, $p(\mathbf{z})$, and $p_{\text{NCP}}(\mathbf{z})$ at different hierarchy/group levels. Samples are projected to a lower dimension ($d = 500$) using PCA and populations are compared via Maximum Mean Discrepancy (MMD). Consistent with Fig. 6.5(a), Tab. 6.9 shows that groups with lower classification loss had a mismatch between $p$ and $q$, and NCP is able to reduce the dissimilarity by re-weighting.

## 6.7 RELATED WORK

In this section, we review related prior works.

[5]ESS measures reliability of SIR via $1/\sum_m (\hat{w}^{(m)})^2$, where $\hat{w}^{(m)} = r(\mathbf{z}^{(m)})/\sum_{m'} r(\mathbf{z}^{(m')})$ (Owen, 2013).

**Energy-based Models (EBMs):** Early work on EBMs for generative learning goes back to 1980s (Ackley et al., 1985; Hinton et al., 1986). Prior to the modern deep learning era, most attempts for building generative models using EBMs were centered around Boltzmann machines (Hinton, 2002; Hinton et al., 2006) and their "deep" extensions (Salakhutdinov & Hinton, 2009; Larochelle & Bengio, 2008). Although the energy function in these models is restricted to simple bilinear functions, they have been proven effective for representing the prior in discrete VAEs (Rolfe, 2016; Vahdat et al., 2018a,b, 2020). Recently, EBMs with neural energy functions have gained popularity for representing complex data distribution (Du & Mordatch, 2019). Pang et al. (2020) have shown that neural EBMs can represent expressive prior distributions. However, in this case, the prior is trained using MCMC sampling, and it has been limited to a single group of latent variables. VAEBM (Xiao et al., 2021) combines VAE's generator with an EBM defined on the pixel space and trains the model using MCMC. Additionally, VAEBM assumes that data lies in a continuous space and applies the energy function in that space. Hence, it cannot be applied to discrete data such as text or graphs. In contrast, NCP-VAE forms the energy function in the latent space and can be applied to non-continuous data. For continuous data, our model can be used along with VAEBM. We believe VAEBM and NCP-VAE are complementary. To eliminate MCMC sampling, NCE (Gutmann & Hyvärinen, 2010) recently is used for training a normalizing flow on data distributions (Gao et al., 2020). Moreover, Han et al. (2019, 2020) use divergence triangulation to sidesteps MCMC sampling. In contrast, we use NCE to train an EBM prior where a noise distribution is easily available through a pre-trained VAE.

**Adversarial Training:** Similar to NCE, generative adversarial networks (GANs) (Goodfellow et al., 2014b) rely on a discriminator to learn the likelihood ratio between noise and real images. However, GANs use the discriminator to update the generator, whereas in NCE, the noise generator is fixed. In spirit similar are recent works (Azadi et al., 2018; Turner et al., 2019; Che et al., 2020) that link GANs, defined in the pixels space, to EBMs. We apply the likelihood ratio trick to the latent space of VAEs. The main difference: the base prior and approximate posterior are trained with the VAE objective rather than the adversarial loss. Adversarial loss has been used for training implicit encoders in VAEs (Makhzani et al., 2015; Mescheder et al., 2017; Engel et al., 2018). But, they have not been linked to energy-based priors as we do explicitly.

**Prior Hole Problem:** Among prior works on this problem, VampPrior (Tomczak & Welling, 2018) uses a mixture of encoders to represent the prior. However, this requires storing training data or pseudo-data to generate samples at test time. Takahashi et al. (2019) use the likelihood ratio estimator to train a simple prior distribution. However at test time, the aggregate posterior is used for sampling in the latent space.

**Reweighted Priors:** Bauer & Mnih (2019) propose a reweighting factor similar to ours, but it is trained via truncated rejection sampling. Lawson et al. (2019) introduce *energy-inspired models (EIMs)* that define distributions induced by the sampling processes used by Bauer & Mnih (2019) as well as our SIR sampling (called SNIS by Lawson et al. (2019)). Although, EIMs have the advantage of end-to-end training, they require multiple samples during training (up to 1K). This can make application of EIMs to deep hierarchical models such as NVAEs very challenging as these models are memory intensive and are trained with a few training samples per GPU. Moreover, our NCP scales easily to hierarchical models where the reweighting factor for each group is trained in parallel with other groups (i.e., NCP enables model parallelism). We view our proposed training method as a simple alternative approach that allows us to scale up EBM priors to large VAEs.

**Two-stage VAEs:** VQ-VAE (Van Den Oord et al., 2017; Razavi et al., 2019) first trains an autoencoder and then fits an autoregressive PixelCNN (Van Den Oord et al., 2016) prior to the latent variables. Albeit impressive results, autoregressive models can be very slow to sample from. Two-stage VAE (2s-VAE) (Dai & Wipf, 2018) trains a VAE on the data, and then, trains another VAE in the latent space. Regularized autoencoders (RAE) (Ghosh et al., 2020) train an autoencoder, and subsequently a Gaussian mixture model on latent codes. In contrast, we train the model with the original VAE objective in the first stage, and we improve the expressivity of the prior using an EBM.

### 6.8 CONCLUSIONS

The prior hole problem is one of the main reasons for VAEs' poor generative quality. In this work, we tackled this problem by introducing the noise contrastive prior (NCP), defined by the product of a reweighting factor and a base prior. We showed how the reweighting factor is trained by contrasting samples from the aggregate posterior with samples from the base prior. Our proposal is simple and can be applied to any VAE to increase its prior's expressivity. We also showed how NCP training scales to large hierarchical VAEs, as it can be done in parallel simultaneously for all the groups. Finally, we demonstrated that NCPs improve the generative performance of small single group VAEs and state-of-the-art NVAEs by a large margin.

# Part IV

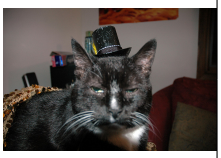
# Conclusion

# CHAPTER 7: CONCLUSION

In the preceding chapters, we have proposed different approaches for generating diverse and accurate outputs for image-captioning and image-generation tasks. In *ConvCap* (Chapter 2), we demonstrate that replacing the LSTM network used in prior image-captioning generators with masked convolutions yields similar performance on standard image-captioning metrics with the additional benefit of faster training since the convolutions are amenable to parallelization. With the goal of training image-captioning models that can describe images in multiple ways, as humans do, we introduced the *PosCap* and *Seq-CVAE* models. In *PosCap* (Chapter 3), we infused diversity by using part-of-speech tags to condition the generation of captions. In *Seq-CVAE* (Chapter 4), we employ latent variable models (VAEs) and design an approach to garner diversity by using hierarchical latent variables for the different words in a caption. In the final work on image captioning, we introduce *DivCap* (Chapter 5), where image-captioning is formulated as a multi-objective optimization problem. We obtain a balance between the accuracy and the diversity of the generated captions. Finally, we proposed *NCP-VAE* (Chapter 6) for high-quality image generation with hierarchical VAEs. This work demonstrates that alleviating the prior-hole problem of VAE's can lead to significant gains in the quality of the generated output. **A significant focus of this thesis has been on designing and learning models that capture the underlying data (text/image) distribution in an efficient manner. This enables us to generate diverse outputs/solutions to the image-generation and image-captioning problems which inherently have multiple plausible solutions.**

# APPENDIX A:

## A.1 QUALITATIVE EXAMPLES - SEQ-CVAE

| Image | Seq-CVAE | POS | AG-CVAE | Div-BS | BS |
|---|---|---|---|---|---|
|  | • a cat sitting on top of a wooden chair<br>• a cat with a collar sits next to a mirror<br>• a cat sitting on top of a table | •a black cat is laying on a couch<br>• a black cat laying on a wooden table<br>• a black cat is laying down on a couch | •a black and white cat sitting on a table.<br>• a black and white cat sitting on a couch.<br>• a black and white cat laying on a chair. | •a black cat laying on top of a wooden table<br>• a black cat laying on top of a couch<br>• a black cat is laying on a couch | •a black cat sitting on top of a pizza<br>• a black cat laying on top of a couch<br>• a black cat laying on top of a pizza |
|  | • a group of birds on some water near water<br>• a couple of birds standing on top of a body of water<br>• two white birds are standing by the water | • a couple of birds that are standing in the water<br>• a couple of birds are standing in the water<br>• two birds standing in a body of water | • a couple of birds are standing in the water.<br>• a couple of birds standing on top of a rock.<br>• a white and white bird standing in the water. | • a couple of birds standing on top of a lake<br>• a couple of birds standing on top of a body of water<br>• a couple of birds that are standing in water | • a couple of birds standing in the water<br>• a couple of birds that are standing in the water<br>• a couple of birds standing on a body of water |
|  | • a giraffe standing in front of a wall<br>• a giraffe standing in front of a wooden fence in a zoo enclosure<br>• a giraffe standing next to a wall with a wall behind | • a giraffe standing next to a building in a zoo<br>• a giraffe standing in front of a building<br>• a baby giraffe is standing in a zoo enclosure | • a couple of giraffe standing next to each other.<br>• a giraffe standing next to a group of giraffes.<br>• a giraffe standing in front of a building. | • a giraffe standing next to a brick wall<br>• a giraffe standing next to a stone wall<br>• a giraffe standing in front of a brick wall | • a giraffe standing next to a brick wall<br>• a giraffe standing next to a stone wall<br>• a giraffe standing next to a wooden fence |
|  | •a bathroom with a toilet and a sink<br>• a bathroom with a toilet and sink in it<br>• a small bathroom with a toilet in the floor | • a white toilet in a small bathroom with a window<br>• a toilet in a bathroom with a window<br>• a white toilet in a bathroom with a toilet | • a bathroom with a toilet and a sink.<br>• a small bathroom with a toilet and a sink.<br>• a small bathroom with a toilet and a window. | • a white toilet sitting in a bathroom next to a wall<br>• a white toilet sitting in a bathroom next to a window<br>• a white toilet sitting in a bathroom next to a door | • a white toilet sitting in a bathroom next to a wall<br>• a white toilet sitting in a bathroom next to a window<br>• a white toilet sitting in a bathroom next to a door |

| Image | Seq-CVAE | POS | AG-CVAE | Div-BS | BS |
|---|---|---|---|---|---|
|  | • a man and a woman standing around a living room<br>• two people standing in a living room playing a game<br>• the two young people are playing a video game | • two people playing a video game in a living room<br>• two people playing a video game in a living room<br>• a young man is playing a video game in a living room | • two men are playing a video game in a living room.<br>• two men playing a video game in a living room.<br>• a group of people in a living room playing wii. | • two people standing in a living room playing a video game<br>• two people playing a video game in a living room<br>• two people standing in a living room playing wii | • two people in a living room playing a video game<br>• two people standing in a living room playing a video game<br>• two people are playing a video game in a living room |
|  | • a glass vase filled with a flower in it<br>• a glass vase is sitting on a table<br>• a glass vase with flowers inside of it | • a small glass vase with some flowers in it<br>• a vase with a bunch of flowers in it<br>• a close up of a vase with flowers in the background | • a glass vase with a vase of flowers in it.<br>• a glass vase with a vase of flowers on it.<br>• a glass vase with a glass of flowers in it. | • a glass vase with some flowers in it<br>• a glass vase filled with lots of flowers<br>• a glass vase with flowers in it next to a window | • a glass vase filled with flowers on a table<br>• a glass vase filled with flowers sitting on a table<br>• a glass vase with flowers in it sitting on a table |
|  | • a white white sink in the bathroom next to<br>• a bathroom with a toilet and a sink<br>• a bathroom with a sink and a toilet | • a very small bathroom with a sink and a mirror<br>• a white bathroom with a sink and mirror<br>• a very small bathroom with a sink and mirror | • a bathroom with a sink and a window.<br>• a bathroom with a sink and a mirror.<br>• a white bathroom with a sink and a mirror. | • a bathroom with a sink and a mirror<br>• a bathroom with a sink and a large mirror<br>• a bathroom with a sink and a mirror in it | • a bathroom with a sink and a mirror<br>• a bathroom with a sink and a window<br>• a white bathroom with a sink and a mirror |
|  | • a street sign and a traffic light on a street<br>• a street with a traffic light on the street<br>• a street light on a city street with traffic | • a stop sign in front of a building with traffic lights<br>• cars are parked on the side of a city street<br>• a red stop sign on a city street | • a city street with a traffic light on it.<br>• an intersection with a street sign and a traffic light.<br>• a street scene with a traffic light on the side. | • a stop sign on the corner of a city street at night<br>• a stop sign on the corner of a city street<br>• a stop sign on the corner of a street next to a traffic | • an intersection with a stop sign and street signs<br>• a stop sign on the corner of a city street<br>• a stop sign on the corner of an empty street |
|  | • a person is surfing in the ocean waves<br>• a couple of people riding a wave in the ocean<br>• a surfer riding a wave while riding a wave | • a man riding a wave on a surfboard in the ocean<br>• a surfer riding a wave in the ocean<br>• a man riding a wave on top of a surfboard | • a man riding a wave on top of a surfboard.<br>• a person riding a wave on top of a surfboard.<br>• a man on a surfboard riding a wave. | • a man riding a wave on top of a surfboard<br>• a man riding a wave on a surfboard<br>• a man riding a wave on a surfboard in the ocean | • a man riding a wave on top of a surfboard<br>• a man riding a wave on a surfboard<br>• a person riding a wave on top of a surfboard |

## A.2 QUALITATIVE EXAMPLES - DIVCAP

Table A.1: **Qualitative examples.** We report captions with worst, best CIDEr score. Besides, we randomly sample one caption to represent average CIDEr. The number following each caption represents the corresponding CIDEr score (best viewed in color).

| Image | LSTM | CVAE | Seq-CVAE | Ours |
|---|---|---|---|---|
|  | a white bench sitting on the side of a white bench (0.38) | an old bench sits in a park (0.35) | a bench with a flower on it sits in a park (0.41) | a bench sitting on top of a park (0.59) |
| | a white bench sitting in front of a garden (1.19) | a bench that is sitting on the ground (0.51) | a bench sitting on a park bench in a park (0.49) | a park bench sitting in front of a building (0.81) |
| | a black and white photo of a bench in a garden (1.20) | a wooden bench sitting in front of a tree (1.44) | a wooden bench sitting in front of a garden (1.79) | a wooden bench sitting in front of a building (1.52) |
|  | a group of elephants crossing a river in a river (0.67) | a group of people riding on top of a large elephant in the water (0.42) | a group of elephants are swimming in the water (0.67) | a group of elephants are in the water of water (0.75) |
| | a group of elephants that are standing in the river (0.91) | a group of people riding elephants in the water (1.45) | a group of elephants walking through water with a man in a water (0.70) | a group of elephants walking in the water (0.98) |
| | a group of people riding elephants in a river (1.61) | a group of people riding elephants in a river (1.61) | a group of people riding elephants on a river (1.62) | a group of elephants walking down a river (1.19) |
|  | a horse standing in the snow next to a snow covered field (0.29) | a dog is standing in a snow covered snow covered field (0.15) | a dog standing next to a small child in a snow covered field (0.17) | a brown and white dog standing on top of a snow (0.23) |
| | a person riding a horse in the snow (0.45) | two dogs are standing in the snow on a snow (1.35) | a couple of animals that are standing in the snow (0.53) | a brown and white dog standing in the snow (0.43) |
| | a dog that is standing in the snow (0.51) | two dogs are standing in the snow (1.67) | a couple of dogs standing in the snow (1.60) | a dog standing in the snow with a person (0.48) |
|  | a couple of men sitting on top of a boat (1.21) | a man and a woman are sitting in the water (0.72) | a man sitting on a boat with a dog on the side of it (1.06) | a man sitting on the top of a boat (1.47) |

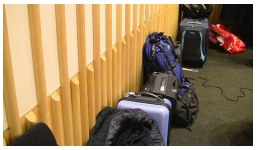
Continued on next page

**Table A.1 – continued from previous page**

| Image | LSTM | CVAE | Seq-CVAE | Ours |
|---|---|---|---|---|
| | two men sitting on a boat in the water (1.57) | a man and a woman are sitting on a boat (1.71) | a man is sitting on a boat with a dog in the background (1.27) | a man sitting on top of a boat on a water (2.41) |
| | a man sitting on a boat in the water (1.84) | a man and woman sitting on the boat on the water (2.21) | a man sitting on a boat on a beach (2.25) | a man is sitting on a boat on the water (2.71) |
|  | a group of skiers standing next to each other (0.14) | a group of people standing in the snow with a snowboard (0.22) | a group of people standing in the snow (0.22) | a group of skis and skis in the snow (0.78) |
| | a group of people standing in the snow (0.22) | several people are standing in the snow on a ski slope (0.69) | a group of people standing on top of a snow covered ski lift (0.72) | a group of skis standing on the snow (0.94) |
| | a group of skis standing on top of a ski slope (1.57) | a group of people standing on top of a snow covered slope (1.26) | a group of people standing on top of a snow covered ski slope (1.07) | a group of skis standing on top of a snow (1.40) |
|  | a man and a woman walking down a city street (0.00) | a man walking down a street with a red umbrella (0.005) | a person riding a skateboard down a street (0.00) | a person walking on the street with a bus (0.009) |
| | a group of people walking down a street (0.44) | people walking down a street at night with umbrellas (0.35) | a woman walking in a city with a bus (0.006) | a man walking on a street at a bus (0.028) |
| | a group of people standing on the side of a street (1.14) | a group of people standing on a street in the rain (1.05) | a group of people on a city street (0.49) | a person is walking down a city street at night (0.25) |
|  | a kitchen filled with lots of clutter and pans (0.71) | a counter with a bunch of food items on it and other items (0.005) | a kitchen with a stove top and a stove top oven (0.40) | a kitchen with pots and pans on the top of the oven (1.64) |
| | a kitchen with lots of pots and pans on top of it (1.44) | a counter with a bunch of food in it (0.005) | a kitchen with lots of pots and pans on it (1.52) | a kitchen with pots and pans on the top of a stove (1.73) |
| | a kitchen with pots and pans on the stove (1.93) | a kitchen with a stove top oven and sink (0.55) | a kitchen with pots and pans on the stove (1.93) | a kitchen with pots and pans on the stove (1.93) |

Continued on next page

**Table A.1 – continued from previous page**

| Image | LSTM | CVAE | Seq-CVAE | Ours |
|---|---|---|---|---|
|  | a person sitting on a bed with a backpack (0.016)<br>a man sitting in a chair with a laptop (0.026)<br>a man sitting on a bed in a room (0.45) | a man sitting on a couch with his arm on a couch in the living (0.007)<br>two people are sitting on a couch in the living room (0.18)<br>a person sitting in the living room with luggage (0.60) | a man sitting on a couch next to a suitcase (0.015)<br>a person sitting on a couch in a room (0.40)<br>a person sitting on a chair in a room (0.41) | a cat sitting on a chair in front of a window (0.020)<br>a cat sitting on the top of a chair (0.024)<br>a cat sitting on top of a chair in a room (0.37) |
|  | a woman and a child are playing a video game (0.011)<br>a man and a woman standing in front of a refrigerator (1.12)<br>a man and a woman standing next to a refrigerator (1.18) | the woman is standing in the kitchen in the kitchen (0.097)<br>a woman standing in front of a refrigerator in the kitchen (0.90)<br>a woman is standing in front of a refrigerator (1.19) | a man and a woman are standing in a kitchen (0.18)<br>a man in a kitchen holding a refrigerator (1.23)<br>a man is standing in front of a refrigerator (1.59) | a woman standing in a refrigerator next to a refrigerator (0.66)<br>a woman standing in front of a refrigerator in a kitchen (0.90)<br>a woman standing in front of a refrigerator (1.11) |
|  | a large brown cow standing on top of a lush ForestGreen hillside (0.001)<br>a large brown cow standing on top of a lush ForestGreen field (0.067)<br>a couple of cows that are standing in a field (1.37) | the cows are standing in the grass near the trees (0.32)<br>a group of cows standing in the grass near a tree (0.35)<br>two cows are standing in a field near some trees (0.66) | a man and a dog standing next to a cow (0.054)<br>two horses are standing in a field near the trees (0.22)<br>a couple of cows that are standing in the dirt (1.34) | a man standing on the top of a brown cow (0.050)<br>a man and a woman walking on the top of a cow (0.091)<br>a man is walking down a dirt field of a cow (0.44) |

## A.3 IMPLEMENTATION DETAILS - NCP-VAE

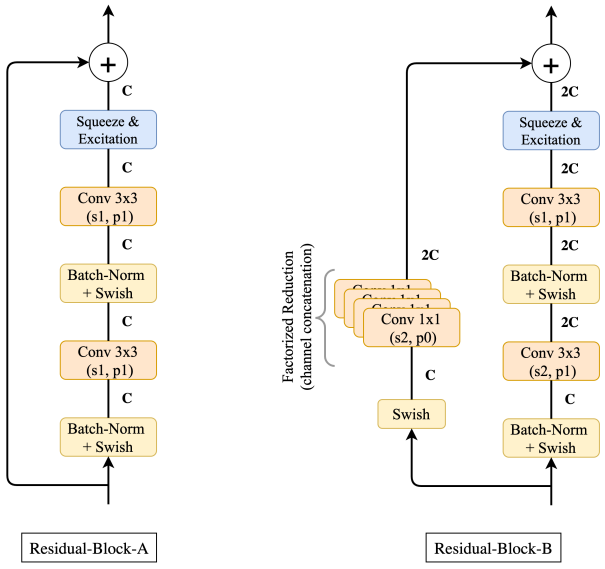
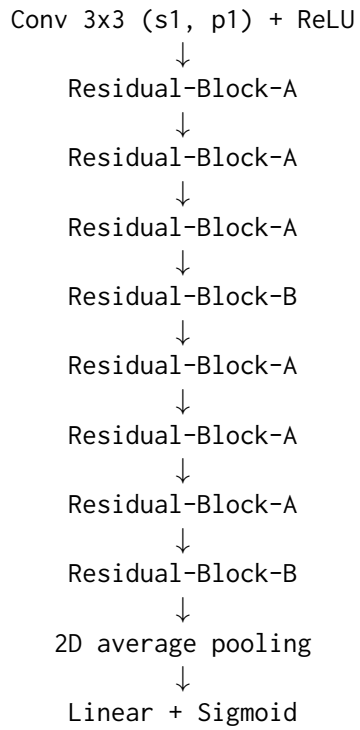



Figure A.1: Residual blocks used in the binary classifier. We use $s$, $p$ and $C$ to refer to the stride parameter, the padding parameter and the number of channels in the feature map, respectively.

The binary classifier is composed of two types of residual blocks as in Fig. A.1. The residual blocks use batch-normalization (Ioffe & Szegedy, 2015), the Swish activation function (Ramachandran et al., 2017), and the Squeeze-and-Excitation (SE) block (Hu et al., 2018). SE performs a *squeeze* operation (*e.g.*, mean) to obtain a single value for each channel. An *excitation* operation (non-linear transformation) is applied to these values to get per-channel weights. The Residual-Block-B differs from Residual-Block-A in that it doubles the number of channels ($C \rightarrow 2C$), while down-sampling the other spatial dimensions. It therefore also includes a factorized reduction with $1 \times 1$ convolutions along the skip-connection. The complete architecture of the classifier is:




| | |
|---|---|
| Optimizer | Adam (Kingma & Ba, 2015) |
| Learning Rate | Initialize at 1$e$-3, CosineAnnealing (Loshchilov & Hutter, 2016) to 1$e$-7 |
| Batch size | 512 (MNIST, CIFAR-10), 256 (CelebA-64), 128 (CelebA HQ 256 ) |

Table A.2: Hyper-parameters for training the binary classifiers.

## A.4 NEAREST NEIGHBORS FROM THE TRAINING DATASET - NCP-VAE

To highlight that hierarchical NCP generates unseen samples at test time rather than memorizing the training dataset, Figures A.2-A.3 visualize samples from the model along with a few training images that are most similar to them (nearest neighbors). To get the similarity score for a pair of images, we downsample to $64 \times 64$, center crop to $40 \times 40$ and compute the Euclidean distance. The KD-tree algorithm is used to fetch the nearest neighbors. We note that the generated samples are quite distinct from the training images.

Query Image Nearest neighbors from the training dataset .

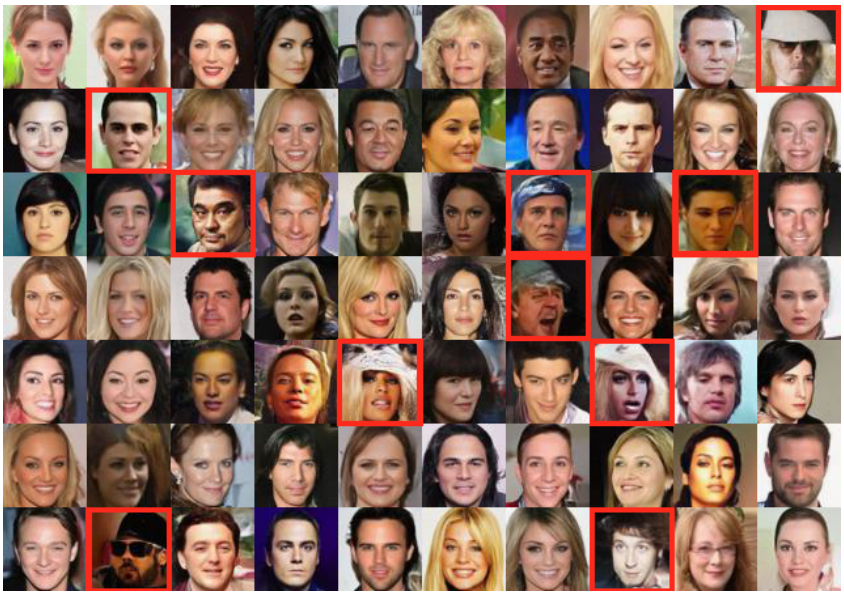
Figure A.2: Query images (left) and their nearest neighbors from the CelebA-HQ-256 training dataset.

Query Image Nearest neighbors from the training dataset .

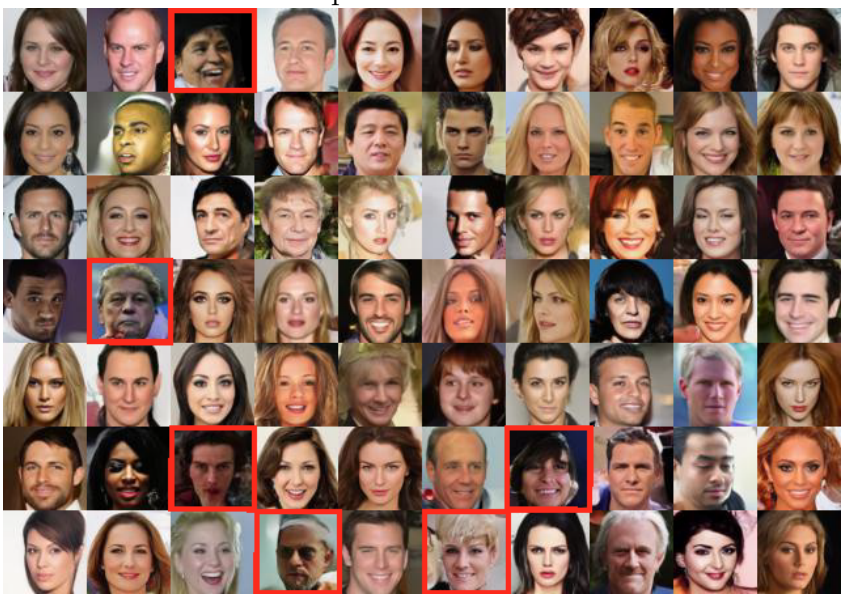
Figure A.3: Query images (left) and their nearest neighbors from the CelebA-HQ-256 training dataset.

## A.5 ADDITIONAL QUALITATIVE EXAMPLES - NCP-VAE

In Fig. A.4, we show additional examples of images generated by NVAE (Vahdat & Kautz, 2020) and our NCP-VAE. We use temperature=0.7 for both. Visually corrupt images are highlighted with a red square.



Random Samples from NVAE at $t = 0.7$



Random Samples from NCP-VAE at $t = 0.7$

Figure A.4: Additional samples from CelebA-64 at $t = 0.7$.

## A.6 ADDITIONAL QUALITATIVE EXAMPLES - NCP-VAE

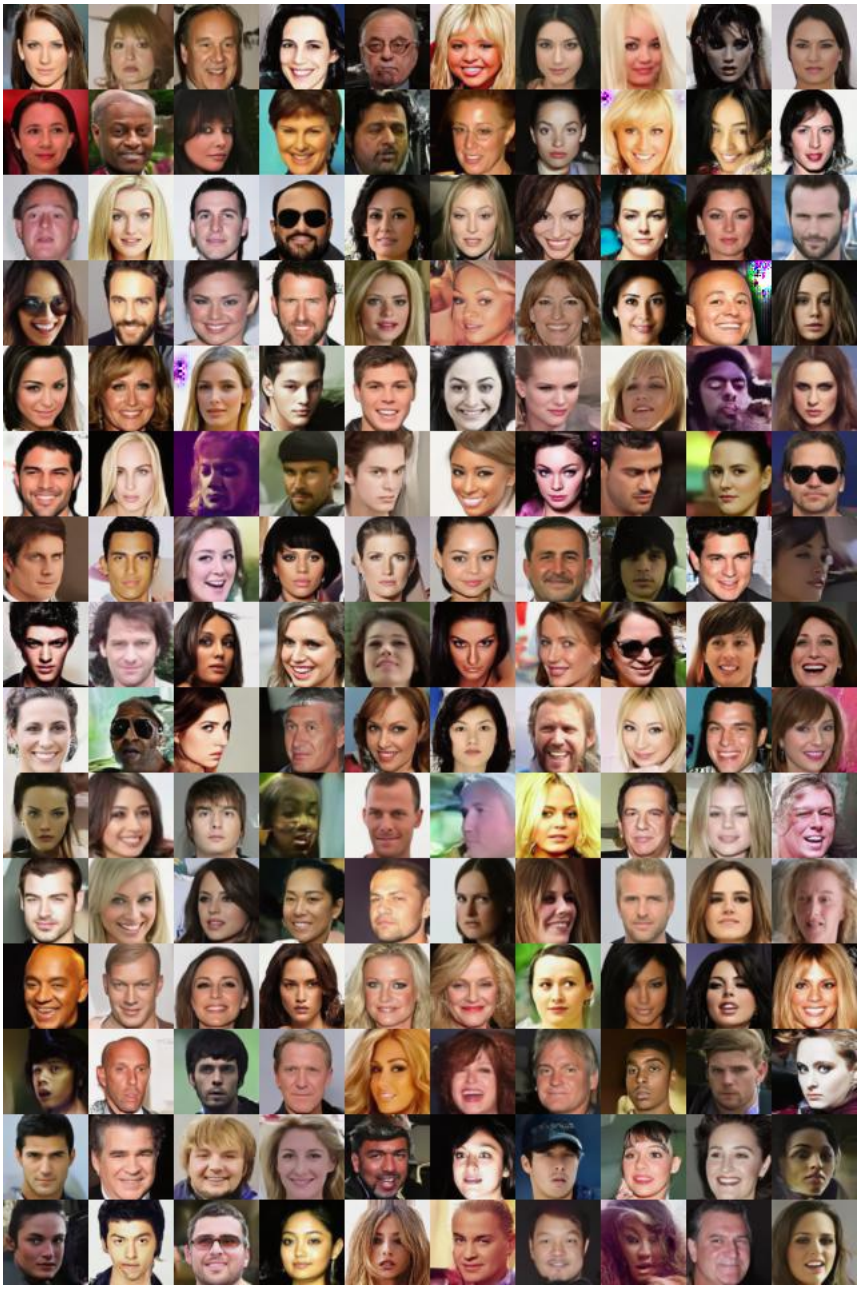


Figure A.5: Additional samples from CelebA-64 at $t = 0.7$.

Figure A.6: Additional samples from CelebA-HQ-256 at $t = 0.7$.

Figure A.7: Selected good quality samples from CelebA-HQ-256.

## REFERENCES


David H Ackley, Geoffrey E Hinton, and Terrence J Sejnowski. A learning algorithm for Boltzmann machines. *Cognitive science*, 9(1):147–169, 1985.

Peter Anderson, Basura Fernando, Mark Johnson, and Stephen Gould. Spice: Semantic propositional image caption evaluation. In *ECCV*, 2016.

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

Jyoti Aneja, Aditya Deshpande, and Alexander G Schwing. Convolutional image captioning. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 5561–5570, 2018.

Jyoti Aneja, Harsh Agrawal, Dhruv Batra, and Alexander Schwing. Sequential latent spaces for modeling the intention during diverse image captioning. In *Proceedings of the IEEE International Conference on Computer Vision*, pp. 4261–4270, 2019.

Jyoti Aneja, Alex Schwing, Jan Kautz, and Arash Vahdat. A contrastive learning approach for training variational autoencoder priors. *Advances in Neural Information Processing Systems*, 34, 2021.

Stanislaw Antol, Aishwarya Agrawal, Jiasen Lu, Margaret Mitchell, Dhruv Batra, C. Lawrence Zitnick, and Devi Parikh. VQA: Visual Question Answering. In *International Conference on Computer Vision (ICCV)*, 2015.

Samaneh Azadi, Catherine Olsson, Trevor Darrell, Ian Goodfellow, and Augustus Odena. Discriminator rejection sampling. In *International Conference on Learning Representations*, 2018.

Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. Neural machine translation by jointly learning to align and translate. *CoRR*, abs/1409.0473, 2014.

Shaojie Bai, J. Zico Kolter, and Vladlen Koltun. An empirical evaluation of generic convolutional and recurrent networks for sequence modeling. *CoRR*, abs/1803.01271, 2018.

Satanjeev Banerjee and Alon Lavie. Meteor: An automatic metric for mt evaluation with improved correlation with human judgments. In *IEEvaluation@ACL*, 2005.

K. Barnard, P. Duygulu, D. Forsyth, N. De Freitas, D. M. Blei, and M. I. Jordan. Matching words and pictures. *JMLR*, 2003.

Matthias Bauer and Andriy Mnih. Resampled priors for variational autoencoders. In *The 22nd International Conference on Artificial Intelligence and Statistics*, pp. 66–75. PMLR, 2019.

Justin Bayer and Christian Osendorfer. Learning stochastic recurrent networks. *arXiv preprint arXiv:1411.7610*, 2014.

Samuel R Bowman, Luke Vilnis, Oriol Vinyals, Andrew Dai, Rafal Jozefowicz, and Samy Bengio. Generating sentences from a continuous space. In *Proceedings of The 20th SIGNLL Conference on Computational Natural Language Learning*, pp. 10–21, 2016.

Andrew Brock, Jeff Donahue, and Karen Simonyan. Large scale gan training for high fidelity natural image synthesis. *arXiv preprint arXiv:1809.11096*, 2018.

Tong Che, Ruixiang Zhang, Jascha Sohl-Dickstein, Hugo Larochelle, Liam Paull, Yuan Cao, and Yoshua Bengio. Your GAN is secretly an energy-based model and you should use discriminator driven latent sampling. *arXiv preprint arXiv:2003.06060*, 2020.

Ciprian Chelba, Tomas Mikolov, Mike Schuster, Qi Ge, Thorsten Brants, Phillipp Koehn, and Tony Robinson. One billion word benchmark for measuring progress in statistical language modeling. *arXiv preprint arXiv:1312.3005*, 2013.

X. Chen and C. L. Zitnick. Mind's eye: A recurrent visual representation for image caption generation. In *2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 2422–2431, June 2015.

Xi Chen, Diederik P Kingma, Tim Salimans, Yan Duan, Prafulla Dhariwal, John Schulman, Ilya Sutskever, and Pieter Abbeel. Variational lossy autoencoder. *arXiv preprint arXiv:1611.02731*, 2016.

Rewon Child. Very deep {vae}s generalize autoregressive models and can outperform them on images. In *International Conference on Learning Representations*, 2021.

K. Cho, A. C. Courville, and Y. Bengio. Describing multimedia content using attention-based encoder-decoder networks. In *IEEE Transactions on Multimedia*, 2015.

Kyunghyun Cho, Bart van Merriënboer, Çağlar Gülçehre, Dzmitry Bahdanau, Fethi Bougares, Holger Schwenk, and Yoshua Bengio. Learning phrase representations using rnn encoder–decoder for statistical machine translation. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pp. 1724–1734, Doha, Qatar, October 2014. Association for Computational Linguistics.

Junyoung Chung, Kyle Kastner, Laurent Dinh, Kratarth Goel, Aaron C Courville, and Yoshua Bengio. A recurrent latent variable model for sequential data. In *Advances in neural information processing systems*, pp. 2980–2988, 2015.

Bin Dai and David Wipf. Diagnosing and enhancing vae models. In *International Conference on Learning Representations*, 2018.

Bo Dai, Sanja Fidler, Raquel Urtasun, and Dahua Lin. Towards diverse and natural image descriptions via a conditional gan. In *Proceedings of the IEEE International Conference on Computer Vision*, pp. 2970–2979, 2017.

Abhishek Das, Satwik Kottur, Khushi Gupta, Avi Singh, Deshraj Yadav, José M. F. Moura, Devi Parikh, and Dhruv Batra. Visual dialog. 2017.

Michael Denkowski and Alon Lavie. Meteor universal: Language specific translation evaluation for any target language. In *Proceedings of the EACL 2014 Workshop on Statistical Machine Translation*, 2014.

Aditya Deshpande, Jyoti Aneja, Liwei Wang, Alexander G Schwing, and David Forsyth. Fast, diverse and accurate image captioning guided by part-of-speech. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 10695–10704, 2019.

Jean-Antoine Désidéri. Multiple-gradient descent algorithm (mgda) for multiobjective optimization. *Comptes Rendus Mathematique*, 350(5-6):313–318, 2012.

Jacob Devlin, Saurabh Gupta, Ross B. Girshick, Margaret Mitchell, and C. Lawrence Zitnick. Exploring nearest neighbor approaches for image captioning. *CoRR*, abs/1505.04467, 2015.

Prafulla Dhariwal, Heewoo Jun, Christine Payne, Jong Wook Kim, Alec Radford, and Ilya Sutskever. Jukebox: A generative model for music. *arXiv preprint arXiv:2005.00341*, 2020.

J. Donahue, L. A. Hendricks, S. Guadarrama, M. Rohrbach, S. Venugopalan, K. Saenko, and T. Darrell. Long-term recurrent convolutional networks for visual recognition and description. In *Proc. CVPR*, 2015.

Yilun Du and Igor Mordatch. Implicit generation and modeling with energy based models. In *Advances in Neural Information Processing Systems*, pp. 3608–3618, 2019.

Jeffrey L Elman. Finding structure in time. *Cognitive science*, 14(2):179–211, 1990.

Jesse Engel, Matthew Hoffman, and Adam Roberts. Latent constraints: Learning to generate conditionally from unconditional generative models. In *International Conference on Learning Representations*, 2018.

H. Fang, S. Gupta, F. Iandola, R. Srivastava, L. Deng, P. Dollár, J. Gao, X. He, M. Mitchell, J. C. Platt, C. L. Zitnick, and G. Zweig. From captions to visual concepts and back. In *Proc. CVPR*, 2015.

A. Farhadi, M. Hejrati, M. A. Sadeghi, P. Young, C. Rashtchian, J. Hockenmaier, and D. Forsyth. Every picture tells a story: Generating sentences from images. In *Proc. ECCV*, pp. 15–29. Springer, 2010.

Jenny Rose Finkel, Christopher D. Manning, and Andrew Y. Ng. Solving the problem of cascading errors: Approximate bayesian inference for linguistic annotation pipelines. In *Proceedings of the 2006 Conference on Empirical Methods in Natural Language Processing*, EMNLP '06, 2006.

Vincent Fortuin, Matthias Hüser, Francesco Locatello, Heiko Strathmann, and Gunnar Rätsch. Som-vae: Interpretable discrete representation learning on time series. In *International Conference on Learning Representations*, 2018.

Marco Fraccaro, Søren Kaae Sønderby, Ulrich Paquet, and Ole Winther. Sequential neural models with stochastic layers. In *Proc. NIPS*, 2016.

Ruiqi Gao, Erik Nijkamp, Diederik P Kingma, Zhen Xu, Andrew M Dai, and Ying Nian Wu. Flow contrastive estimation of energy-based models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 7518–7528, 2020.

Jonas Gehring, Michael Auli, David Grangier, Denis Yarats, and Yann N. Dauphin. Convolutional sequence to sequence learning. *CoRR*, 2017.

Partha Ghosh, Mehdi S. M. Sajjadi, Antonio Vergari, Michael Black, and Bernhard Scholkopf. From variational to deterministic autoencoders. In *International Conference on Learning Representations*, 2020.

Kevin Gimpel, Dhruv Batra, Chris Dyer, and Gregory Shakhnarovich. A systematic exploration of diversity in machine translation. In *In Proc. of EMNLP*, 2013.

Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative Adversarial Networks. In *Proc. NIPS*, 2014a.

Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial nets. In Z. Ghahramani, M. Welling, C. Cortes, N. Lawrence, and K. Q. Weinberger (eds.), *Advances in Neural Information Processing Systems*, volume 27, pp. 2672–2680. Curran Associates, Inc., 2014b. URL https://proceedings.neurips.cc/paper/2014/file/5ca3e9b122f61f8f06494c97b1afccf3-Paper.pdf.

Anirudh Goyal ALIAS PARTH Goyal, Alessandro Sordoni, Marc-Alexandre Côté, Nan Rosemary Ke, and Yoshua Bengio. Z-forcing: Training stochastic recurrent networks. In *Proc. NIPS*, 2017.

Karol Gregor, Frederic Besse, Danilo Jimenez Rezende, Ivo Danihelka, and Daan Wierstra. Towards conceptual compression. In *Advances In Neural Information Processing Systems*, pp. 3549–3557, 2016.

Ishaan Gulrajani, Kundan Kumar, Faruk Ahmed, Adrien Ali Taiga, Francesco Visin, David Vazquez, and Aaron Courville. PixelVAE: A latent variable model for natural images. *arXiv preprint arXiv:1611.05013*, 2016.

Michael Gutmann and Aapo Hyvärinen. Noise-contrastive estimation: A new estimation principle for unnormalized statistical models. In *Proceedings of the Thirteenth International Conference on Artificial Intelligence and Statistics*, pp. 297–304, 2010.

Tian Han, Erik Nijkamp, Xiaolin Fang, Mitch Hill, Song-Chun Zhu, and Ying Nian Wu. Divergence triangle for joint training of generator model, energy-based model, and inferential model. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 8670–8679, 2019.

Tian Han, Erik Nijkamp, Linqi Zhou, Bo Pang, Song-Chun Zhu, and Ying Nian Wu. Joint training of variational auto-encoder and latent energy-based model. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 7978–7987, 2020.

Martin Heusel, Hubert Ramsauer, Thomas Unterthiner, Bernhard Nessler, and Sepp Hochreiter. Gans trained by a two time-scale update rule converge to a local nash equilibrium. In *Advances in neural information processing systems*, pp. 6626–6637, 2017.

Geoffrey E Hinton. Training products of experts by minimizing contrastive divergence. *Neural computation*, 14(8):1771–1800, 2002.

Geoffrey E Hinton, Terrence J Sejnowski, et al. Learning and relearning in boltzmann machines. *Parallel distributed processing: Explorations in the microstructure of cognition*, 1(282-317):2, 1986.

Geoffrey E Hinton, Simon Osindero, and Yee-Whye Teh. A fast learning algorithm for deep belief nets. *Neural computation*, 18(7):1527–1554, 2006.

Jonathan Ho, Ajay Jain, and Pieter Abbeel. Denoising diffusion probabilistic models. *arXiv preprint arxiv:2006.11239*, 2020.

S. Hochreiter and J. Schmidhuber. Long short-term memory. *Neural Computation*, 1997a.

Sepp Hochreiter and Jürgen Schmidhuber. Long short-term memory. *Neural Comput.*, 9(8): 1735–1780, November 1997b. ISSN 0899-7667.

Micah Hodosh, Peter Young, and Julia Hockenmaier. Framing image description as a ranking task: Data, models and evaluation metrics. *J. Artif. Int. Res.*, 47(1), May 2013. ISSN 1076-9757.

Matthew D Hoffman and Matthew J Johnson. Elbo surgery: yet another way to carve up the variational evidence lower bound. In *Workshop in Advances in Approximate Bayesian Inference, NIPS*, 2016.

Jie Hu, Li Shen, and Gang Sun. Squeeze-and-excitation networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 7132–7141, 2018.

Junjie Hu, Yu Cheng, Zhe Gan, Jingjing Liu, Jianfeng Gao, and Graham Neubig. What makes a good story? designing composite rewards for visual storytelling. In *Thirty-Fourth AAAI Conference on Artificial Intelligence (AAAI)*, 2020.

Y.-T. Hu, J.-B. Huang, and A. G. Schwing. MaskRNN: Instance Level Video Object Segmentation. In *Proc. NIPS*, 2017.

Sergey Ioffe and Christian Szegedy. Batch normalization: Accelerating deep network training by reducing internal covariate shift. In *International Conference on Machine Learning*, pp. 448–456, 2015.

Pavel Izmailov, Polina Kirichenko, Marc Finzi, and Andrew Gordon Wilson. Semi-supervised learning with normalizing flows. In *ICML*, 2020.

U. Jain, S. Lazebnik, and A. G. Schwing. Two can play this Game: Visual Dialog with Discriminative Question Generation and Answering. In *Proc. CVPR*, 2018.

Unnat Jain, Ziyu Zhang, and Alexander G. Schwing. Creativity: Generating diverse questions using variational autoencoders. In *Computer Vision and Pattern Recognition*, 2017.

Eric Jang, Shixiang Gu, and Ben Poole. Categorical reparameterization with gumbel-softmax. 2017.

Yangqing Jia, Mathieu Salzmann, and Trevor Darrell. Learning cross-modality similarity for multinomial data. In *Proceedings of the 2011 International Conference on Computer Vision*, ICCV '11, pp. 2407–2414, Washington, DC, USA, 2011. IEEE Computer Society.

J. Johnson, A. Karpathy, and L. Fei-Fei. DenseCap: Fully Convolutional Localization Networks for Dense Captioning. In *Proc. CVPR*, 2016.

A. Karpathy and L. Fei-Fei. Deep visual-semantic alignments for generating image descriptions. In *2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2015.

Tero Karras, Timo Aila, Samuli Laine, and Jaakko Lehtinen. Progressive growing of gans for improved quality, stability, and variation. *arXiv preprint arXiv:1710.10196*, 2017.

Tero Karras, Samuli Laine, and Timo Aila. A style-based generator architecture for generative adversarial networks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 4401–4410, 2019.

Tero Karras, Miika Aittala, Janne Hellsten, Samuli Laine, Jaakko Lehtinen, and Timo Aila. Training generative adversarial networks with limited data. *arXiv preprint arXiv:2006.06676*, 2020.

D Kingma and J Ba. Adam: A method for stochastic optimization in: Proceedings of international conference on learning representations. 2015.

D. P. Kingma and M. Welling. Auto-Encoding Variational Bayes. In *ICLR*, 2014.

D. P. Kingma, D. J. Rezende, S. Mohamed, and M. Welling. Semi-Supervised Learning with Deep Generative Models. In *Proc. NIPS*, 2014.

Diederik P Kingma, Tim Salimans, Rafal Jozefowicz, Xi Chen, Ilya Sutskever, and Max Welling. Improved variational inference with inverse autoregressive flow. In *Advances in Neural Information Processing Systems*, pp. 4743–4751, 2016.

Durk P Kingma and Prafulla Dhariwal. Glow: Generative flow with invertible 1x1 convolutions. In S. Bengio, H. Wallach, H. Larochelle, K. Grauman, N. Cesa-Bianchi, and R. Garnett (eds.), *Advances in Neural Information Processing Systems 31*, pp. 10236–10245, 2018.

R. Kiros, R. Salakhutdinov, and R. S. Zemel. Unifying visual-semantic embeddings with multimodal neural language models. In *TACL*, 2015.

Alexej Klushyn, Nutan Chen, Richard Kurle, Botond Cseke, and Patrick van der Smagt. Learning hierarchical priors in vaes. In *Advances in Neural Information Processing Systems*, 2019.

Alex Krizhevsky, Geoffrey Hinton, et al. Learning multiple layers of features from tiny images, 2009.

G. Kulkarni, V. Premraj, S. Dhar, S. Li, Y. Choi, A. C. Berg, and T. L. Berg. Baby talk: Understanding and generating simple image descriptions. In *Proc. CVPR*, 2011.

Hugo Larochelle and Yoshua Bengio. Classification using discriminative restricted boltzmann machines. In *Proceedings of the 25th international conference on Machine learning*, pp. 536–543, 2008.

John Lawson, George Tucker, Bo Dai, and Rajesh Ranganath. Energy-inspired models: Learning with sampler-induced distributions. In *NeurIPS*, 2019.

Yann LeCun. The mnist database of handwritten digits. *http://yann. lecun. com/exdb/mnist/*, 1998.

Dianqi Li, Xiaodong He, Qiuyuan Huang, Ming-Ting Sun, and Lei Zhang. Generating diverse and accurate visual captions by comparative adversarial learning. *arXiv preprint arXiv:1804.00861*, 2018.

Chieh Hubert Lin, Chia-Che Chang, Yu-Sheng Chen, Da-Cheng Juan, Wei Wei, and Hwann-Tzong Chen. Coco-gan: generation by parts via conditional coordinating. In *Proceedings of the IEEE International Conference on Computer Vision*, pp. 4512–4521, 2019.

Chin-Yew Lin. Rouge: a package for automatic evaluation of summaries. July 2004.

Chin-Yew Lin and Eduard H. Hovy. Automatic evaluation of summaries using n-gram co-occurrence statistics. In *HLT-NAACL*, 2003.

Tsung-Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C Lawrence Zitnick. Microsoft coco: Common objects in context. In *Proc. ECCV*, 2014.

Qiang Liu and Dilin Wang. Stein variational gradient descent: A general purpose bayesian inference algorithm. In *Advances in neural information processing systems*, pp. 2378–2386, 2016.

Siqi Liu, Zhenhai Zhu, Ning Ye, Sergio Guadarrama, and Kevin Murphy. Improved image captioning via policy gradient optimization of spider. In *Proceedings of the IEEE international conference on computer vision*, pp. 873–881, 2017.

Ziwei Liu, Ping Luo, Xiaogang Wang, and Xiaoou Tang. Deep learning face attributes in the wild. In *Proceedings of the IEEE international conference on computer vision*, pp. 3730–3738, 2015.

Ilya Loshchilov and Frank Hutter. Sgdr: Stochastic gradient descent with warm restarts. *arXiv preprint arXiv:1608.03983*, 2016.

Jiasen Lu, Jianwei Yang, Dhruv Batra, and Devi Parikh. Neural baby talk, 2018.

Ruotian Luo and Gregory Shakhnarovich. Analysis of diversity-accuracy tradeoff in image captioning. *ArXiv*, abs/2002.11848, 2020.

Ruotian Luo, Brian Price, Scott Cohen, and Gregory Shakhnarovich. Discriminability objective for training descriptive captions. *arXiv preprint arXiv:1803.04376*, 2018.

Lars Maaløe, Marco Fraccaro, Valentin Liévin, and Ole Winther. BIVA: A very deep hierarchy of latent variables for generative modeling. In *Advances in neural information processing systems*, pp. 6548–6558, 2019.

Pranava Madhyastha, Josiah Wang, and Lucia Specia. Vifidel: Evaluating the visual fidelity of image descriptions. *arXiv preprint arXiv:1907.09340*, 2019.

Alireza Makhzani, Jonathon Shlens, Navdeep Jaitly, Ian Goodfellow, and Brendan Frey. Adversarial autoencoders. *arXiv preprint arXiv:1511.05644*, 2015.

Junhua Mao, Wei Xu, Yi Yang, Jiang Wang, Zhiheng Huang, and Alan Yuille. Deep Captioning with Multimodal Recurrent Neural Networks (m-rnn). In *Proc. ICLR*, 2015.

Lars Mescheder, S Nowozin, and Andreas Geiger. Adversarial variational bayes: Unifying variational autoencoders and generative adversarial networks. In *34th International Conference on Machine Learning (ICML)*, pp. 2391–2400. PMLR, 2017.

Nasrin Mostafazadeh, Ishan Misra, Jacob Devlin, Margaret Mitchell, Xiaodong He, and Lucy Vanderwende. Generating natural questions about an image. In *ACL (1)*. The Association for Computer Linguistics, 2016.

M. Mostajabi, P. Yadollahpour, and G. Shakhnarovich. Feedforward semantic segmentation with zoom-out features. In *2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 3376–3385, June 2015.

Charlie Nash and Conor Durkan. Autoregressive energy machines. *arXiv preprint arXiv:1904.05626*, 2019.

Aaron van den Oord, Sander Dieleman, Heiga Zen, Karen Simonyan, Oriol Vinyals, Alex Graves, Nal Kalchbrenner, Andrew Senior, and Koray Kavukcuoglu. Wavenet: A generative model for raw audio. *arXiv preprint arXiv:1609.03499*, 2016.

Aaron van den Oord, Yazhe Li, and Oriol Vinyals. Representation learning with contrastive predictive coding. *arXiv preprint arXiv:1807.03748*, 2018.

Vicente Ordonez, Girish Kulkarni, and Tamara L Berg. Im2text: Describing images using 1 million captioned photographs. In *Proceedings of the 24th International Conference on Neural Information Processing Systems*, NIPS'11, 2011.

Art B. Owen. *Monte Carlo theory, methods and examples.* 2013.

Bo Pang, Tian Han, Erik Nijkamp, Song-Chun Zhu, and Ying Nian Wu. Learning latent space energy-based prior model. *arXiv preprint arXiv:2006.08205*, 2020.

Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. Bleu: a method for automatic evaluation of machine translation. In *ACL*, 2001.

Kancharla Parimala and Sumohana Channappayya. Quality aware generative adversarial networks. In *Advances in Neural Information Processing Systems*, pp. 2948–2958, 2019.

Razvan Pascanu, Tomas Mikolov, and Yoshua Bengio. On the difficulty of training recurrent neural networks. In *Proceedings of the 30th International Conference on International Conference on Machine Learning - Volume 28*, ICML'13, pp. III–1310–III–1318. JMLR.org, 2013.

Matthew E. Peters, Mark Neumann, Mohit Iyyer, Matt Gardner, Christopher Clark, Kenton Lee, and Luke Zettlemoyer. Deep contextualized word representations. In *Proc. of NAACL*, 2018.

Stanislav Pidhorskyi, Donald A Adjeroh, and Gianfranco Doretto. Adversarial latent autoencoders. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 14104–14113, 2020.

Wei Ping, Kainan Peng, Kexin Zhao, and Zhao Song. Waveflow: A compact flow-based model for raw audio. *ICML*, 2020.

Yunchen Pu, Zhe Gan, Ricardo Henao, Xin Yuan, Chunyuan Li, Andrew Stevens, and Lawrence Carin. Variational autoencoder for deep learning of images, labels and captions. In *NIPS*, 2016.

Prajit Ramachandran, Barret Zoph, and Quoc V Le. Searching for activation functions. *arXiv preprint arXiv:1710.05941*, 2017.

Marc'Aurelio Ranzato, Sumit Chopra, Michael Auli, and Wojciech Zaremba. Sequence level training with recurrent neural networks. *arXiv preprint arXiv:1511.06732*, 2015.

Ali Razavi, Aaron van den Oord, and Oriol Vinyals. Generating diverse high-fidelity images with vq-vae-2. In *Advances in Neural Information Processing Systems*, pp. 14837–14847, 2019.

Joseph Redmon, Santosh Kumar Divvala, Ross B. Girshick, and Ali Farhadi. You only look once: Unified, real-time object detection. *CoRR*, abs/1506.02640, 2015.

Shaoqing Ren, Kaiming He, Ross Girshick, and Jian Sun. Faster R-CNN: Towards real-time object detection with region proposal networks. In *Neural Information Processing Systems (NIPS)*, 2015.

Steven J. Rennie, Etienne Marcheret, Youssef Mroueh, Jarret Ross, and Vaibhava Goel. Self-critical sequence training for image captioning. *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 1179–1195, 2017a.

Steven J Rennie, Etienne Marcheret, Youssef Mroueh, Jarret Ross, and Vaibhava Goel. Self-critical sequence training for image captioning. In *CVPR*, pp. 7008–7024, 2017b.

Danilo Jimenez Rezende and Fabio Viola. Taming vaes. *arXiv preprint arXiv:1810.00597*, 2018.

Danilo Jimenez Rezende, Shakir Mohamed, and Daan Wierstra. Stochastic backpropagation and approximate inference in deep generative models. In *International Conference on Machine Learning*, pp. 1278–1286, 2014.

Jason Tyler Rolfe. Discrete variational autoencoders. *arXiv preprint arXiv:1609.02200*, 2016.

Mihaela Rosca, Balaji Lakshminarayanan, and Shakir Mohamed. Distribution matching in variational inference. *arXiv preprint arXiv:1802.06847*, 2018.

Olga Russakovsky, Jia Deng, Hao Su, Jonathan Krause, Sanjeev Satheesh, Sean Ma, Zhiheng Huang, Andrej Karpathy, Aditya Khosla, Michael Bernstein, Alexander C. Berg, and Li Fei-Fei. Imagenet large scale visual recognition challenge. *International Journal of Computer Vision*, 115(3):211–252, Dec 2015. ISSN 1573-1405.

Ruslan Salakhutdinov and Geoffrey Hinton. Deep boltzmann machines. In *Artificial intelligence and statistics*, pp. 448–455, 2009.

I. Schwartz, A. G. Schwing, and T. Hazan. High-Order Attention Models for Visual Question Answering. In *Proc. NIPS*, 2017.

Ozan Sener and Vladlen Koltun. Multi-task learning as multi-objective optimization. In *Advances in Neural Information Processing Systems*, pp. 527–538, 2018.

Evan Shelhamer, Jonathan Long, and Trevor Darrell. Fully convolutional networks for semantic segmentation. *IEEE Trans. Pattern Anal. Mach. Intell.*, 39(4):640–651, April 2017. ISSN 0162-8828. doi: 10.1109/TPAMI.2016.2572683.

Rakshith Shetty, Marcus Rohrbach, Lisa Anne Hendricks, Mario Fritz, and Bernt Schiele. Speaking the same language: Matching machine to human captions by adversarial training. In *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, 2017.

Kevin J. Shih, Saurabh Singh, and Derek Hoiem. Where to look: Focus regions for visual question answering. In *Computer Vision and Pattern Recognition*, 2016.

K. Simonyan and A. Zisserman. Very deep convolutional networks for large-scale image recognition. In *ICLR*, 2015.

Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*, 2014.

R. Socher, A. Karpathy, Q. V. Le, C. D. Manning, and A. Y. Ng. Grounded compositional semantics for finding and de- scribing images with sentences. In *Proc. TACL*, 2014a.

Richard Socher, Andrej Karpathy, Quoc Le, Christopher Manning, and Andrew Ng. Grounded compositional semantics for finding and describing images with sentences. *Transactions of the Association for Computational Linguistics*, 2:207–218, 2014b.

K. Sohn, X. Yan, and H. Lee. Learning Structured Output Representation using Deep Conditional Generative Models. In *Proc. NIPS*, 2015.

Casper Kaae Sønderby, Tapani Raiko, Lars Maaløe, Søren Kaae Sønderby, and Ole Winther. Ladder variational autoencoders. In *Advances in neural information processing systems*, pp. 3738–3746, 2016.

Yang Song, Jascha Sohl-Dickstein, Diederik P Kingma, Abhishek Kumar, Stefano Ermon, and Ben Poole. Score-based generative modeling through stochastic differential equations. In *International Conference on Learning Representations*, 2021.

Ilya Sutskever, James Martens, and Geoffrey Hinton. Generating text with recurrent neural networks. In Lise Getoor and Tobias Scheffer (eds.), *Proceedings of the 28th International Conference on Machine Learning (ICML-11)*, ICML '11, pp. 1017–1024, New York, NY, USA, June 2011. ACM. ISBN 978-1-4503-0619-5.

Hiroshi Takahashi, Tomoharu Iwata, Yuki Yamanaka, Masanori Yamada, and Satoshi Yagi. Variational autoencoder with implicit optimal priors. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 33, pp. 5066–5073, 2019.

I Tolstikhin, O Bousquet, S Gelly, and B Schölkopf. Wasserstein auto-encoders. In *International Conference on Learning Representations (ICLR 2018)*. OpenReview. net, 2018.

Jakub Tomczak and Max Welling. Vae with a vampprior. In *International Conference on Artificial Intelligence and Statistics*, pp. 1214–1223, 2018.

Ryan Turner, Jane Hung, Eric Frank, Yunus Saatchi, and Jason Yosinski. Metropolis-hastings generative adversarial networks. In *International Conference on Machine Learning*, pp. 6345–6353. PMLR, 2019.

Arash Vahdat and Jan Kautz. NVAE: A deep hierarchical variational autoencoder. In *Neural Information Processing Systems (NeurIPS)*, 2020.

Arash Vahdat, Evgeny Andriyash, and William G Macready. DVAE#: Discrete variational autoencoders with relaxed Boltzmann priors. In *Neural Information Processing Systems*, 2018a.

Arash Vahdat, William G. Macready, Zhengbing Bian, Amir Khoshaman, and Evgeny Andriyash. DVAE++: Discrete variational autoencoders with overlapping transformations. In *International Conference on Machine Learning (ICML)*, 2018b.

Arash Vahdat, Evgeny Andriyash, and William G Macready. Undirected graphical models as approximate posteriors. In *International Conference on Machine Learning (ICML)*, 2020.

Aäron Van Den Oord, Nal Kalchbrenner, and Koray Kavukcuoglu. Pixel recurrent neural networks. In *Proceedings of the 33rd International Conference on International Conference on Machine Learning*, pp. 1747–1756. JMLR. org, 2016.

Aäron van den Oord, Nal Kalchbrenner, Oriol Vinyals, Lasse Espeholt, Alex Graves, and Koray Kavukcuoglu. Conditional image generation with pixelcnn decoders. In *NIPS*, 2016.

Aaron Van Den Oord, Oriol Vinyals, et al. Neural discrete representation learning. In *Advances in Neural Information Processing Systems*, pp. 6306–6315, 2017.

L. van der Maaten and G. E. Hinton. Visualizing Data Using t-SNE. *JMLR*, 2008.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. Attention is all you need. *CoRR*, 2017.

Ramakrishna Vedantam, C Lawrence Zitnick, and Devi Parikh. Cider: Consensus-based image description evaluation. In *Proc. CVPR*, 2015.

Ashwin K. Vijayakumar, Michael Cogswell, Ramprasaath R. Selvaraju, Qing Sun, Stefan Lee, David J. Crandall, and Dhruv Batra. Diverse beam search for improved description of complex scenes. In *Proc. AAAI*, 2018.

O. Vinyals, A. Toshev, S. Bengio, and D. Erhan. Show and tell: A neural image caption generator. In *Proc. CVPR*, 2015a.

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

Josiah Wang, Pranava Swaroop Madhyastha, and Lucia Specia. Object counts! bringing explicit detections back into image captioning. In *Proceedings of the North American Chapter of the Association of Computational Linguistics: Human Language Technologies (NAACL HLT)*. Association for Computational Linguistics, 2018.

L. Wang, A. G. Schwing, and S. Lazebnik. Diverse and Accurate Image Description Using a Variational Auto-Encoder with an Additive Gaussian Encoding Space. In *Proc. NIPS*, 2017a.

Liwei Wang, Yin Li, and Svetlana Lazebnik. Learning two-branch neural networks for image-text matching tasks. *CoRR*, abs/1704.03470, 2017b.

Liwei Wang, Alexander Schwing, and Svetlana Lazebnik. Diverse and accurate image description using a variational auto-encoder with an additive gaussian encoding space. In *Advances in Neural Information Processing Systems*, pp. 5756–5766, 2017c.

Liwei Wang, Alexander G. Schwing, and Svetlana Lazebnik. Diverse and accurate image description using a variational auto-encoder with an additive gaussian encoding space. In *Proc. NIPS*, 2017d.

Qingzhong Wang and Antoni B. Chan. Cnn+cnn: Convolutional decoders for image captioning. *CoRR*, abs/1805.09019, 2018.

Qingzhong Wang and Antoni B. Chan. Describing like humans: On diversity in image captioning. *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 4190–4198, 2019.

Ronald J Williams. Simple statistical gradient-following algorithms for connectionist reinforcement learning. In *Reinforcement Learning*, pp. 5–32. Springer, 1992.

Zhisheng Xiao, Karsten Kreis, Jan Kautz, and Arash Vahdat. {VAEBM}: A symbiosis between variational autoencoders and energy-based models. In *International Conference on Learning Representations*, 2021.

Haowen Xu, Wenxiao Chen, Jinlin Lai, Zhihan Li, Youjian Zhao, and Dan Pei. On the necessity and effectiveness of learning the prior of variational auto-encoder. *arXiv preprint arXiv:1905.13452*, 2019.

Kelvin Xu, Jimmy Ba, Ryan Kiros, Kyunghyun Cho, Aaron Courville, Ruslan Salakhudinov, Rich Zemel, and Yoshua Bengio. Show, attend and tell: Neural image caption generation with visual attention. In *International conference on machine learning*, pp. 2048–2057, 2015.

Yezhou Yang, Ching Lik Teo, Hal Daumé, III, and Yiannis Aloimonos. Corpus-guided sentence generation of natural images. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, EMNLP '11, pp. 444–454, Stroudsburg, PA, USA, 2011. Association for Computational Linguistics.

Ting Yao, Yingwei Pan, Yehao Li, Zhaofan Qiu, and Tao Mei. Boosting image captioning with attributes. In *IEEE International Conference on Computer Vision, ICCV 2017, Venice, Italy, October 22-29, 2017*, 2017a.

Ting Yao, Yingwei Pan, Yehao Li, Zhaofan Qiu, and Tao Mei. Boosting image captioning with attributes. In *Proceedings of the IEEE International Conference on Computer Vision*, pp. 4894–4902, 2017b.

Mark Yatskar, Luke Zettlemoyer, and Ali Farhadi. Situation recognition: Visual semantic role labeling for image understanding. In *Conference on Computer Vision and Pattern Recognition*, 2016.

R. A. Yeh, J. Xiong, W.-M. Hwu, M. Do, and A. G. Schwing. Interpretable and Globally Optimal Prediction for Textual Grounding using Image Concepts. In *Proc. NIPS*, 2017.

Zijun Zhang, Ruixiang Zhang, Zongpeng Li, Yoshua Bengio, and Liam Paull. Perceptual generative autoencoders. In *International Conference on Machine Learning*, 2020.