Multimodal Data Augmentation for Image Captioning

Multimodale Datenanreicherung für Bilduntertitelung
Master thesis by Oliver Hahn
Date of submission: March 31, 2022

1. Review: Prof. Stefan Roth, Ph.D.
2. Review: M.Sc. Shweta Mahajan
Darmstadt

TECHNISCHE UNIVERSITÄT DARMSTADT

Department of Computer Science
Visual Inference Lab

Erklärung zur Abschlussarbeit gemäß §22 Abs. 7 und §23 Abs. 7 APB der TU Darmstadt

Hiermit versichere ich, Oliver Hahn, die vorliegende Masterarbeit ohne Hilfe Dritter und nur mit den angegebenen Quellen und Hilfsmitteln angefertigt zu haben. Alle Stellen, die Quellen entnommen wurden, sind als solche kenntlich gemacht worden. Diese Arbeit hat in gleicher oder ähnlicher Form noch keiner Prüfungsbehörde vorgelegen.

Mir ist bekannt, dass im Fall eines Plagiats (§38 Abs. 2 APB) ein Täuschungsversuch vorliegt, der dazu führt, dass die Arbeit mit 5,0 bewertet und damit ein Prüfungsversuch verbraucht wird. Abschlussarbeiten dürfen nur einmal wiederholt werden.

Bei der abgegebenen Thesis stimmen die schriftliche und die zur Archivierung eingereichte elektronische Fassung gemäß §23 Abs. 7 APB überein.

Bei einer Thesis des Fachbereichs Architektur entspricht die eingereichte elektronische Fassung dem vorgestellten Modell und den vorgelegten Plänen.

Darmstadt, March 31, 2022

O. Hahn

Abstract

Image captioning is an important application combining vision and language models with the aim to automatically generate a textual description for a given image. To overcome the limited annotated multimodal image-text data used for training, recent work gathers contextual descriptions of different samples describing similar image contexts. Building upon this idea of separating the image-text data further into objects and contexts, a multimodal data augmentation method for image captioning is developed. Leveraging basic computer vision and natural language processing techniques as well as a multimodal pre-training model leads to a lightweight method for augmentation of both modalities. New meaningful data is obtained by swapping specific image sections and caption tokens of an image-text pair with those of a contentwise reasonable second image-text pair. Experiments leading to architecture choices of the method are presented and a detailed analysis of the obtained data is carried out. Furthermore, it is examined how the additional augmented training data affects the captioning performance of an image captioning as well as a diverse image captioning framework. The evaluation is performed on the domain used for training and augmentation as well as on a second domain in order to assess the generalization capabilities.

Zusammenfassung

Automatische Bilduntertitelung vereint Modelle des Maschinellen Sehens und der Linguistischen Datenverarbeitung mit dem Ziel automatisiert ein gegebenes Bild in Form von Text zu beschreiben. Um die begrenzte Verfügbarkeit annotierter gepaarter Bild-Text Daten zu umgehen, werden in der aktuellen Forschung kontextuelle Beschreibungen verschiedener Bilder, welche einen ähnlichen kontext abbilden gesammelt. Aufbauend auf dieser Idee der Aufteilung der Bild-Text-Daten in Objekte und Kontexte wird im Rahmen dieser Arbeit eine Methode zur Anreicherung multimodaler Daten für Bilduntertitelung entwickelt. Aus der Kombination grundlegender Techniken des Maschinellen Sehens und der Linguistischen Datenverarbeitung mit einem multimodalen pre-training Model geht schließlich eine effiziente Methode zur Augmentierung beider Modalitäten hervor. Neue aussagekräftige Daten werden durch das Vertauschen bestimmter Bildausschnitte sowie der entsprechenden Wörter in der Bilduntertitelung eines Bild-Text-Paares mit denen eines inhaltlich sinnvollen zweiten Bild-Text-Paares erzeugt. Es werden Experimente vorgestellt, anhand deren die Architektur der Methode erarbeitet wurde. Des Weiteren wird eine detaillierte Analyse der mittels der Augmentierungsmethode erzeugten Daten durchgeführt. Darüber hinaus wird untersucht, wie sich die zusätzlichen Trainingsdaten auf Methoden zur automatisierten Erzeugung von Bilduntertitelungen auswirken. Dies wird einerseits in Bezug auf ein Model zur direkten Erzeugung einer Bildunterschrift untersucht. Zum anderen, wird dies in Hinsicht auf ein Model zur Erzeugung mehrerer diverser Bildunterschriften wiederholt. Die Evaluierung erfolgt zum einen auf der zum Training und der Augmentierung verwendeten Domäne, zum anderen wird die Generalisierungseigenschaft mithilfe einer zweiten Domäne untersucht.

Contents

List of Figures

List of Tables

1 Introduction

1.1 Motivation

In today's modern world we are constantly surrounded by visual and textual data. In fact, we spend parts of our lives in a digital world that consists almost exclusively of images and language data in some form. Such multimodal data can be understood and interpreted very well by humans. Since the integration of artificial intelligence into all areas of life is progressing steadily, the handling of multimodal data is being intensively researched. One of the most relevant tasks in this context is image captioning which refers to the automated generation of a natural language description for a given image. This constitutes a highly challenging task in the field of generative intelligence and scene understanding as it requires to combine computer vision (CV) and natural language processing (NLP). The potential of image captioning goes far beyond applications analyzing large amounts of visual data [43], such as in social media. For example, image captioning applications are capable of generating assistance diagnoses in the medical field [84] or enabling visually impaired people access to the visual domain of the real world and the internet [2, 98].
Over the last years, image captioning research led to a variety of proposed methods and considerable progress in the field of image captioning. First deterministic approaches [98, 40, 4] can generate one sentence per image and achieve near human syntactic and semantic properties. Nevertheless, it is often not possible to represent the content of an image in only one sentence. Diverse image captioning models [5, 59] attempt to overcome this limitation and generate a variety of diverse natural language captions for a single image. Inter alia, this progress results from the continuous efforts in creating high quality multimodal datasets [56, 47, 1] which provide large amounts of diverse content captured in images along with high quality human annotated textual descriptions. However, this is an extremely cost and time intensive process as human assistance can hardly be circumvented and in some cases even highly skilled experts are needed for annotation. Given that, a limitation arises in the development of better image captioning methods as deep learning models steadily become larger and accordingly the amount of training data needed to access the full potential increases. Furthermore it is the accepted notion that larger amounts of data result in better deep learning models through regularization effects, reduced overfitting and increased generalization capabilities, as showed in [90].
Data augmentation addresses this fundamental problem of modern deep learning approaches by increasing the diversity of training data without explicitly collecting new data. This translates into enriching the dataset with augmented data samples, which are created by manipulation or generation based on the original data. Data augmentation is therefore a cheap way to acquire larger amounts of training data and thus extract more information from the expensive annotated data. This approach aligns well with the general objective of machine learning to maximize the generalization capability in order to reduce the amount annotated data. In the context of image captioning, augmentation especially multimodal data augmentation is comparatively unexplored. To the best of our knowledge there exists no method to augment both the visual and textual modality of paired image-text data in a meaningful way, which leads to the assumption that the full potential of multimodal datasets has not yet been exploited.

1.2 Goals and Contribution

Generally, it can be assumed that the entity of a multimodal dataset contains information beyond these accessible by directly learning the paired data as it is typical practice in captioning methods. Despite considerable progress in image captioning, little research has been done on multimodal augmentation. Therefore, this research is dedicated to the task of exploring the potential of multimodal data augmentation for image captioning.
Hereby this work is building up on Mahajan and Roth [59], proposing a diverse captioning method that can generate highly accurate as well as diverse image captions and leverages contextual descriptions all over the dataset describing similar contexts in different visual scenes. Such a strong and diverse model provides an ideal challenge when developing an augmentation method. In addition, experiments are carried out on a widely used standard image captioning model by Anderson et al. [4]. In contrast to previous work, this research aims to develop a data augmentation method for image captioning that augments both text and image modalities of a paired image-text dataset. Furthermore, the aim is to develop a lightweight and applicable method and provide valuable insights into multimodal data augmentation.

1.3 Outline

Starting with the initial idea of leveraging a lightweight text-guided generative approach for image manipulation, the course of this work leads to the finally presented method, which combines basic techniques from CV and NLP with a multimodal pre-training approach. A variety of experiments is then conducted to analyze and evaluate the proposed method.

Necessary foundations are discussed in Chapter 2. Hereby, the relevant neural network architectures are addressed in Sec. 2.1. Next, the related work regarding image captioning and diverse image captioning is presented in Sec. 2.2. Followed by an introduction to the field of image manipulation 2.3 and data augmentation 2.4.
Chapter 3 aims to present the method developed in this thesis. First the initial idea of augmentation utilizing generative adversarial networks to manipulate images is discussed in 3.1. Afterwards, the developed CutSwap multimodal augmentation technique is presented in detail in Sec. 3.2.
Chapter 4 describes the overall framework of this research regarding implementation details 4.1, the used datasets 4.2 and metrics 4.3. Furthermore, components of the method are investigated experimentally in Sec. 4.4. Sections 4.5 and 4.6 describe the analysis of the obtained augmented data as well as investigations regarding the impact of the additional augmented training data on the performance of captioning approaches. Finally, the conducted research is summarized and an outlook for future research is given in Chapter 5.

2 Related Work

Image captioning has the objective of generating a textual description for a given image and thus connects CV and NLP. In this context, the following work deals with augmenting the multimodal data used to train captioning models. In addition to captioning, this work builds on a variety of tasks and methods, such as image generation and manipulation, data augmentation as well as basic CV and NLP techniques. The following section serves to lay these foundations. Firstly relevant neural network architectures are introduced before shifting focus to image captioning as well as image manipulation. Moreover, an introduction to data augmentation and existing applications for image captioning is given.

2.1 Neural Networks

Breaking it down, the image captioning process can be divided into understanding the image and generating the corresponding textual description. In order to obtain an effective representation of the image content, it is common to use dense features from convolutional neural networks (CNN) [50] originally used for image classification or object detection. To subsequently generate a text description from the extracted image features Recurrent Neural Networks (RNN) [78] are widely used but also generative models like Variational Autoencoders (VAE) [44] can be used as language models. These networks and their relevant representatives are introduced within the following section. In addition, Generative Adversarial Networks (GAN) [32], which play a major role in the context of image augmentation and manipulation, are discussed thereafter. Furthermore, an introduction to Transformer Networks [95], which have recently became widely popular, especially in NLP and multimodal learning, is given.

2.1.1 Convolutional Neural Networks

After being proposed by LeCun et al. [50] and others, CNNs mainly gained popularity due to the performance of AlexNet [48] in the ImageNet Large Scale Visual Recognition Challenge [79]. The network is the first approach successfully using convolutional and pooling layers for feature extraction as well as fully connected layers to produce a distribution over given classes. VGG Net [87] improves upon this by using smaller convolution filters and a significantly deeper architecture. Szegedy et al. [92] present GoogleNet, which improves performance as well as efficiency. The introduced inception module utilizes very small convolutions combined with pooling operations and concatenation of the respective outputs to reduce the amount of parameters. He et al. [35] propose a new architecture that includes skip connections between layers as well as strong batch normalization. Residual Neural Networks (ResNet) allow to train very deep networks with better performance and less complexity. Leading to a paradigm shift towards improving model accuracy without increasing complexity. While the research focus at the time when these networks were designed was on image

classification, it is now on more complex tasks. Nevertheless, such networks are still used as a component of modern methods and serve, *e. g.* as a backbone.

2.1.2 Recurrent Neural Networks

Working with language, a general objective of a model is to specify the probability of certain words appearing together within a sequence. This is the underlying component of many NLP tasks, providing the model with the ability to stochastically deal with natural language. Based on a vocabulary and conditioned with the encoded image X, the language model auto-regressively predicts a sequence of words y width length n.

$$P\big(y_1, y_2, ...y_n|X\big) = \prod_{i=1}^{n} P\big(y_i|y_1, y_2, ...y_{i-1}|X\big) \tag{2.1}$$

RNNs are intended for usage with sequentially related data by preserving information from the past in order to perform future predictions by utilizing a internal hidden state. Proposed by Rumelhart et al. [78] the vanilla RNN uses a feedback loop which updates the hidden state at a timestep t based on the input x and the previous hidden state h_{t-1}. Given an activation function f and trainable weights W, constant for each timestep, the prediction y_t results as follows:

$$\begin{aligned} h_t &= f\big(W_{hh}h_{h-t} + W_{xh}x_t\big) \\ y_t &= W_{hy}h_t \end{aligned} \tag{2.2}$$

A difficulty that arises in the context of vanilla RNNs is the vanishing and exploding gradient problem [69]. This results from the circumstance that capturing long-term dependencies the multiplicative gradient can result in an exponential decrease or increase with respect to the network depth. While exploding gradients can be easily avoided by gradient clipping, there is a variety of proposed architectures that resolve the vanishing gradient problem when capturing long-term dependencies.

Long Short-Term Memory Networks The Long Short-Term Memory (LSTM) [37] is a widely known architecture when dealing with the vanishing gradient problem, and one of the predominant options for language modeling. This architecture is designed to improve gradient flow properties by leveraging a second hidden state, called cell state c_t. Moreover, it is characterized by gates controlling the input to c_t and the impact of the input signal x_t and c_t on the hidden state h_t. Similar to the vanilla RNN, the hidden state of the previous timestep, x_t and trainable weights W are used to compute four different gates. The input gate i controlling the information flow into c_t, the forget gate f which can be interpreted as controlling the amount of dropped information from c_{t-1}, as well as the output gate o controlling the amount of information of c_t revealed to h_t and gate g controlling the information kept in c. With the help of Sigmoid and Hyperbolic Tangent activation functions and the element-wise product, the hidden state in timestep t is obtained as follows:

$$\begin{pmatrix} i_t \\ f_t \\ o_t \\ g_t \end{pmatrix} = \begin{pmatrix} \sigma \\ \sigma \\ \sigma \\ \tanh \end{pmatrix} W \begin{pmatrix} h_{t-1} \\ x_t \end{pmatrix}$$
$$c_t = f_t \odot c_{t-1} + i_t \odot g_t \tag{2.3}$$
$$h_t = o_t \odot \tanh\left(c_t\right)$$
$$y_t = W_{hy} h_t$$

In a language model, this is used in combination with a softmax-function to obtain the probability distribution over the given vocabulary of size v in every timestep t.

$$P_t(h_{ti}, W_{hy}) = \frac{\exp W_{hy} h_{ti}}{\sum_j^v \exp W_{hy} h_{tj}} \tag{2.4}$$

Decoding Recurrent Predictions RNN language models predict a probability distribution over the entire vocabulary for each token in the output sequence. Hence, a decoder process is needed to transform these probabilities into a final sequence of words. Since the set of possible solutions is the vocabulary size exponentiated by the length of the output sequence, it is intractable to evaluate all possibilities. The simplest and fastest approach is to greedily choose the sub-sequence with the highest probability in each timestep. However, this leads to weak results. Instead of considering every timestep in an isolated manner, Beam Search [91] keeps multiple best sub-sequences for every step depending on the beam width and considers all combinations of preceding sub-sequences with the current. This leads to improved results at the cost of computational overhead.

2.1.3 Variational Autoencoder

Besides the previously presented models, standard generative approaches such as VAE can be used for language modeling. These models are building up on the concept of Auto-Encoders which learn an encoder to compress data into a significantly smaller latent space and a decoder to reconstruct it again. However, these models are disadvantageous as they do not allow to sample from the latent space in order to generate novel data.
To circumvent this, Kingma and Welling [44] propose the VAE as a way to model probabilistic data generation. Assuming that the training data X is generated from some underlying observed latent representations $p(X|z)$ by sampling from a true prior $z \sim p(z)$ followed by sampling from the true conditional distribution $X \sim p(X|z)$. Since it is intractable to directly optimise on this, an additional encoder model is introduced which is modeling $q_\theta(z|x)$ to approximate the distribution $p_\Theta(z|x)$ modelled by the decoder network. These probabilistic encoder $q_\theta(z|x)$ and decoder $p_\Theta(x|z)$ networks produce distributions over z and x respectively which are chosen to be Gaussian $q(z|X) = \mathcal{N}(\mu_\phi(x_i), \Sigma_\phi(x_i))$. Where μ_ϕ and Σ_ϕ are arbitrary deterministic functions and ϕ are the trainable parameters. With the help of some minor assumptions, the approximation is fitted to $p(z|X)$ by minimizing the Kullback-Leibler divergence (KLD). This finally results in the lower bound for $p(X|z)$ as:

$$\log\ p(X) \geq \mathbb{E}_{q_\phi(z|X)}\big[\log p_\theta(X|z)\big] - D_{KL}\big[q_\phi(z|X), p(z)\big] \tag{2.5}$$

In order to train the described model it requires one additional modification since the minimization of the KLD as well as the maximization of the log-likelihood using stochastic gradient descend (SGD) is not possible as sampling from $p(X|z)$ is not differentiable. The re-parameterization trick overcomes this issue by sampling from a random noise $\epsilon \sim \mathcal{N}(0,1)$ and retrieving the latent vector trough $z = \mu_\phi(x_i) + \sqrt{\Sigma_\phi}(x_i) \cdot \epsilon$ in which only μ and Σ are learned. This is differentiable since the sampling process is outside the gradient flow. In order to generate new data during inference time, the encoder is no longer needed instead one samples directly from the prior $p(z)$
The conditional variational auto-encoder (CVAE) [88] extends the previously examined model in the sense that both encoder and decoder are provided with a conditioning input. This leads to a structured latent space by conditioning all the distributions on the variable c. The modified variational lower bound of the CVAE follows as:

$$\log\ p(X|c) \geq \mathbb{E}_{q_\phi(z|X,c)}\big[\log\ p_\theta(X|z,c)\big] - D_{KL}\big[q_\phi(z|X,c)||p(z|c)\big] \tag{2.6}$$

Overall, this allows a directed control on the data generation process.

2.1.4 Generative Adversarial Networks

While VAEs are frequently used when dealing with natural language as these models yielding strong performance. In contrast maximizing a lower bound of likelihood leads to qualitatively poor and blurred generations when dealing with image data. For the purpose of generating examples from a high dimensional training distribution, Goodfellow et al. [32] derived the idea of GANs. The Vanilla GAN is a game theoretical approach of learning to transform samples of a simple distribution to the high dimensional training distribution through a two-player game. One player being a generator network G trying to generate new data points from random noise z with statistics similar to the training data distribution p_{data}. The second player being a discriminator model D trying to distinguish whether a sample is drawn from the training data or generated by G. Both networks are jointly trained in a min-max game in which D wants to maximize the objective and G aims to minimize the objective:

$$\min_{\theta_g}\max_{\theta_d}\Big[\mathbb{E}_{x\sim p_{data}}\log D_{\theta_d}(x) + \mathbb{E}_{z\sim p(z)}\log(1 - D_{\theta_d}(G_{\theta_g}(z)))\Big] \tag{2.7}$$

The training is performed through alternating between gradient ascent on the discriminator network and gradient descent on the generator network. Unfortunately, the definition $\log(1 - D_{\theta_d}(G_{\theta_g}(z)))$ yields the problem that the gradient signal of the generator network is dominated by samples for which the networks already performs well. This is resolved by slightly modifying the objective. Instead of minimizing the likelihood of the discriminator being correct, the likelihood of it being wrong is maximized. This, in turn, leads to a higher gradient signal for bad samples:

$$\max_{\theta_g}\mathbb{E}_{z\sim p(z)}\log(D_{\theta_d}(G_{\theta_g}(z))) \tag{2.8}$$

Once the network is trained only to generator is needed when generating new data from random noise z during inference time.
An influential work based the previously described Vanilla GAN is the deep convolutional generative adversarial network (DCGAN) by Radford et al. [73]. DCGAN is an architecture based on convolutional operations. This

is realized by replacing pooling layers with strided convolutions in the discriminator network and fractional-strided convolutions in the generator. Moreover all fully connected hidden layers are removed. This design gives both the generator and discriminator an advanced spatial reasoning ability which leads to a significant improvement in image quality compared to the Vanilla GAN. In recent years, there has been intensive research on GANs with a focus on improving the training properties [6], applying GANs to a wide variety of tasks while creating a multitude of novel architectures [62, 114, 11, 39, 41]. The most influential works in this field are the following. Conditional GAN [62] enables conditioned data generation by feeding an auxiliary information into both generator and discriminator. Wasserstein GAN [6] proposes an algorithm to improve the training procedure. Cycle-GAN [114] leverages two GANs as well as inverse mapping for the task of image-to-image translation in the absence of paired examples. StyleGAN [11] uses an alternative generator architecture borrowing from style transfer in order to enable intuitive and scale-specific control of the synthesis. Brock et al. [11] propose BigGAN, a large-scale model yielding state-of-the-art realistic looking results in image generation. The current trend in GAN research tends towards the usage of transformer networks like in recently proposed TransGAN [39]. While modern GANs perform remarkably well in generating natural looking images, they suffer from the disadvantage of being dependent on large amounts of computing resources. For example, a StyleGAN3 [41] requires to be trained on eight NVIDIA Tesla V100 GPUs to generate 1024×1024 resolution images.

2.1.5 Transformer Networks

In sequence processing, RNN architectures have the disadvantage of not being parallelizable and loosing information over long distances. CNNs, on the other hand, are parallelizable but have the disadvantage that the processable sequence length is limited by the receptive field. Self-attention [8] overcomes both disadvantages, as it is highly parallelizable and good at handling long sequences. This approach became widely popular since Vaswani et al. [95] proposed the Transformer network. A new architecture building up on self-attention while showing superior performance in dealing with sequence-to-sequence tasks. The vanilla transformer consists of a encoder and a decoder part. Both parts consist of a stack of identical building blocks. Within the encoder, each block is mainly composed out of a multi-head self-attention module and a position-wise feed-forward network. In addition, residual connections [35] are used to allow for a deeper architecture. Given values V and keys K being the encoded source sequence and the query Q being the output of the target sentence and the encoding size d_k, attention results as follows:

$$\text{Attention}\big(Q, K, V\big) = \text{softmax}\Big(\frac{QK^T}{\sqrt{d_k}}\Big)V \tag{2.9}$$

Instead of applying this directly, multi-head attention is used which linearly projects Q, K and V multiple times using different learned linear projections and applies the attention function afterwards. The final output then results from concatenating and projecting once again.
The transformer model led to the emerge of a new paradigm in NLP. Hereby, large transformer models are pre-trained on huge amounts of data and afterwards fine-tuned on a specific downstream task. Following this idea Devlin et al. [22] propose Bidirectional Encoder Representations from Transformers (BERT) and achieve state-of-the-art results on multiple downstream tasks. Hereby the bidirectional multi-layer transformer model is pre-trained on two self-supervised tasks. The first task is called masked language modelling, where the model needs to predict a masked token in a sequence. The second task focuses on sentence prediction. Here the model receives a pair of sentences as input and tries to predict whether the second sentence is a subsequent sentence to the first. A similar approach is the generative pre-trained transformer [12]. Within

this approach, a model with 175 billion parameters is trained in a self-supervised manner with the objective in predicting the next word of a sequence given the previous context tokens. This results in state-of-the-art or on-par zero-shot performance on multiple downstream tasks without being fine-tuned on the tasks nor the data.

2.2 Image Captioning

Image captioning is the task of generating a natural language description for the content of an image. Therefore, a model has to understand the visual information and be capable of generating meaningful and syntactically correct sentences. In a simplified way, all deep learning-based solutions encode the input image into one or multiple features which serve as the input for a language model that then generates a sequence of words. By now, there is a large number of proposed methods for both visual encoding as well as language modeling. Visual encoding can be divided into non-attentive methods based on global CNN features, additive attentive methods embedding on the basis of a grid or regions, graph-based methods adding relations between visual regions, and self-attentive methods operating on a transformer basis. Language models in image captioning can be divided into LSTM-based models, CNN-based models, Transformer-based models, and image-text early fusion models.
One of the first approaches in deep-learning based image captioning is the work conducted by Vinyals et al. [98] and Karpathy and Li [40]. Both methods extract high-level, fixed-sized representations leveraging the top layer activations in a CNN. These are used as a conditioning element to a LSTM-based language model. Vinyals et al. [98] combine the output of a pre-trained GoogleNet with a LSTM. The CNN maps the image into the same space as the word embeddings. In an initial step, the image and start token are fed into the LSTM, based on which the LSTM predicts the first token. Successively, each next token is predicted on the basis of the previous prediction until the prediction of the stop word indicates that a complete sequence has been generated. In addition to the LSTM, the top layer of the CNN and word embeddings are all trained simultaneously. Karpathy and Li [40] follows a similar approach, but focuses more on the multimodal commonality of the data. The presented approach aligns sentence parts to corresponding image regions utilizing multimodal embeddings. These correspondences serve as training data for the actual captioning model. Further differences to [98] are that a pre-trained AlexNet [48] is used for visual encoding and the retrieved features are inserted as a bias term at the first step of the text generation. The advantage of these models is their simplicity and the compactness of the image representations. Based on this, the visual information can be condensed over the entire image context. However, this also yields the disadvantage of lacking capability to generate fine-grained captions. Anderson et al. [4] integrate attention into the visual encoding process to overcome this drawback. The proposed method combines a bottom-up and top-down visual attention mechanism. Bottom-up is realized by utilizing Faster R-CNN [77] to predict salient image regions and represent these by a pooled convolutional feature vector. Faster R-CNN is a object detection model aiming to identify to detect object instances of certain classes. Hereby a lightweight Region Proposal Network predicts object proposals at a intermediate level of a CNN. For each spatial location a bounding box as well as class-agnostic objectness score is predicted. In a following step, pooling of the region of interest is applied to extract a feature vector for each proposal. Building up on this, non-maximum suppression (NMS) for each object class is applied. The image features $V = \{v_1, ...v_k\}$ are the mean-pooled convolutional features of the respective region. This results in a ”hard” attention as only few image bounding box features are used regardless of the training objective. Another element of the method is a pre-training on the Visual Genome [47] dataset, which introduces an auxiliary loss by learning to predict attributes to the object classes. This leads to dense and rich feature representations and therefore became a widely used approach for image captioning. The proposed captioning architecture

consists of two LSTM-layers. Given the features V "soft" top-down attention is applied by the first LSTM layer to weight each feature when generating the caption. The second LSTM is the actual language model. At each step the attention LSTM is fed the output of the language LSTM of the previous timestep concatenated with the mean-pooled image feature $\bar{v}$ and the encoded previously generated word $W_e \Pi_t$. Given the output h_t^1 of the attention LSTM a normalized attention weight for each image feature is generated. Moreover a convex combination of all image features is calculated as input to the language model. The following Equation 2.10 shows this process, here W_{va}, W_{hs} and w_a are learnable parameters.

$$\begin{aligned} \alpha_{i,t} &= w_a^T tanh(W_{va} v_i + W_{ha} h_t^1) \\ \alpha_t &= softmax(\alpha_t) \\ \hat{v}_t &= \sum_{i=1}^{K} \alpha_{i,t} v_i \end{aligned} \tag{2.10}$$

The input of the language LSTM consist of the attended image feature $\hat{v}_t$, concatenated with the output of the attention LSTM h_t^1. This finally leads to the conditioned distribution over possible output words for every timestep. The architecture of the captioning model is visualized in Figure 2.1.

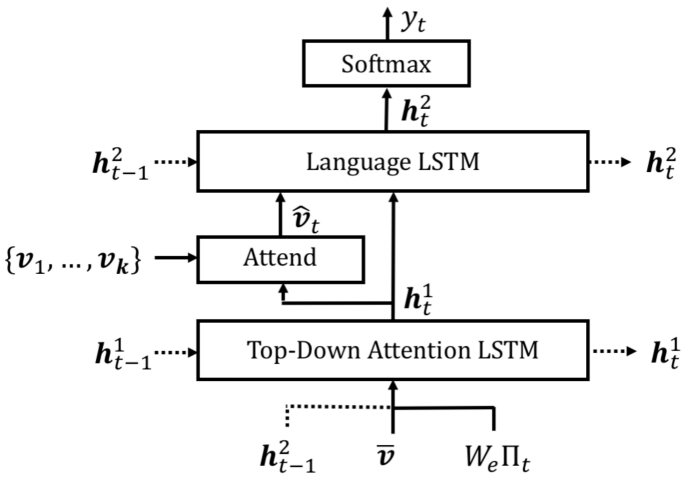


Figure 2.1: UpDown captioning model architecture [4].

Many existing image captioning methods work similarly and mainly introduce minor novelties, *e. g.* changing the attention mechanism of the visual encoding or modifying the architecture and type of the language model. Thereby, research shifts focus towards Transformer [95] networks for language modeling. X-linear attention [67] proposes a technique to strengthen the representative capacity of the output attended feature. This is realized by simultaneously taking advantage of the spatial and channel-wise bi-linear attention distributions to capture the second-order interactions between the input features. This results in enhanced region-level and image-level features. Further upcoming approaches are early fusion [54] and vision-and-language pre-training [57] models. OSCAR proposed by Li et al. [54] and VinVL proposed by Zhang et al. [111] are *BERT*-like architectures characterized by being pre-trained on large amounts of data as well as the fact that visual and textual data are mixed at a very early stage of the architecture.

Diverse Image Captioning The previously described methods optimize for single, accurate image captions which makes them incapable of modeling one-to-many relationships. Diverse captioning aims to overcome this issue by creating models that can replicate the quality and variability of human captioning. First works in this area try to generate diverse captions by different sampling methods in a high dimensional space. Vijayakumar et al. [97] modify vanilla beam search by dividing the beams into similar groups and encouraging diversity between these groupings using word-to-word distance. Since this is fairly limited, other approaches try to introduce diversity by using a contrastive learning approach [20] and generative models like conditional GAN architectures [19]. Unfortunately, the performance of these models is quite restricted in terms of captioning quality, so that recent research shifted focus towards VAEs [5, 59].
The image captioning method by Mahajan and Roth [59] is such an approach and serves as the main model used in this work. The context-object split conditional variational auto-encoder (COS-CVAE) learns a CVAE conditioned on images to sample diverse captions. In order to learn beyond the underlying information of the paired data, the method leverages contextual descriptions of the dataset that describe similar visual scenes. The method utilizes a novel factorization of the latent space into a context and object part. Due to this, the ground-truth caption is split into $x = \{x^m, x^o\}$, being a list of objects from the visual scene described in the caption x^o, and the contextual description x^m, being the caption excluding the object tokens. By utilizing [28], contextual embeddings are learned and used to retrieve nearest neighboring contextual descriptions. During training these additional contextual descriptions are composed to a pseudo ground-truth caption by inserting the object information from the image for the masked out tokens enabling context-based pseudo supervision through an auxiliary loss-term. Given a image-caption pair, the training input image features are extracted following Anderson et al. [4] and averaged in a first step. x^m is extracted from the caption by replacing the occurring object with a placeholder and x^o is simply a list containing these objects. The model consists of two LSTMs modeling the posterior distributions q_{θ^o} and q_{θ^m} in sequential latent spaces. During the training process the averaged image features $\bar{v}$ and x^m are the input to LSTM q_{θ^m} resulting in the encoding of the textual information z^m. Consecutively, x^o, z^m and $\bar{v}$ serve as the input to LSTM q_{θ^o} to encode the object information z^o. Similarly, the conditional Gaussian distribution priors p_{ϕ^m} and p_{ϕ^o} are modeled using LSTMs taking $\bar{v}$ and in the latter case $\bar{v}$ and z^m as input. In order to model the posterior distribution the attention LSTM of Anderson et al. [4] is employed. The hidden state of the language LSTM at the previous timestep, the context vector and the encoded representation $[z_t, \bar{v}]$ as well as the ground-truth word of the caption at timestep $t-1$ serve as the input to the attention LSTM. The output of the attention LSTM and the attended image features are the input to the language LSTM to generate the token at timestep t. Given the image I, the latent context representation $\tilde{z}^m$ is sampled from the prior p_{ϕ^m} before sampling the conditional sequential prior p_{ϕ^o} using $\tilde{z}^m$ and I. The latent representations are then input to the attention LSTM predicting the caption. Both the training and inference procedures are visualized in Figure 2.2.
This novel structure of the latent space as well as leveraging the contextual descriptions over the dataset lead to state-of-the-art performance in diverse image captioning.

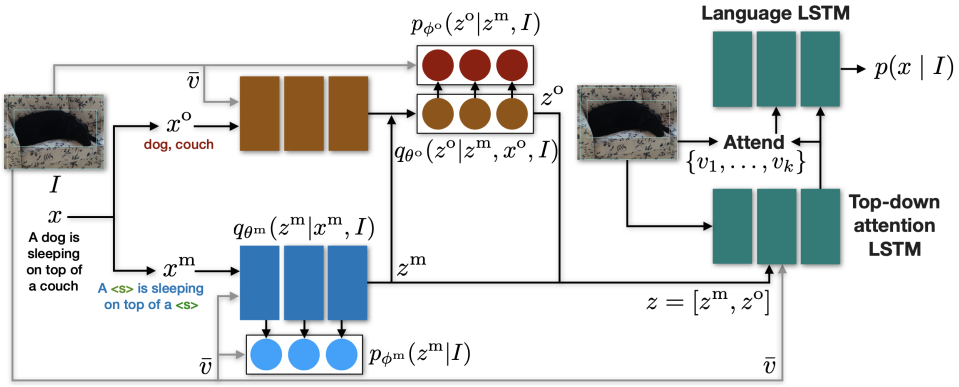


Figure 2.2: COS-CVAE model architecture [59].

2.3 Image Manipulation

Image manipulation aims to manipulate certain parts or properties of a given image and is closely related to image generation. Image generation refers to the task of generating new data that yields the properties of previously learned data. Generally, such tasks can be performed conditioned or unconditioned. Extended to a multimodal level, this can mean generating an image based on the condition being a text sequence. This refers to text-to-image generation. In this multimodal context image manipulation aims to manipulate parts of an image based on a textual prompt. As one of the pioneering works in text-guided image generation, Reed et al. [76] propose a method utilizing a conditional GAN [62] trained on text embeddings by a previously trained encoder network instead of using class labels. This leads to visually appealing images based on the textual descriptions, especially in case of less complex data containing mainly one class [99, 66]. Zhang et al. [109] improve image quality by decomposing the generation process into two consecutive steps using a multi-scale GAN architecture. Hereby, one GAN is generating primitive shapes and colours based on the given text description while the second one is generating the high resolution output image from the text description and the low resolution image. AttnGAN [104] further improves the image quality by introducing a deep attentional multimodal similarity model to compute a fine-grained image-text matching loss for training a multi-stage generator.
Based on these approaches, more and more research emerged utilizing text prompts for image manipulation. Dong et al. [25] propose a CNN based encoder-decoder architecture in which the encoded text description is concatenated with the image features and decoded to obtain the manipulated image. Nam et al. [63] introduce a text-adaptive discriminator providing fine-grained word-level training feedback. Based on the AttnGAN [104] architecture, Li et al. [51] propose a multi-stage network that uses a novel image-text combination module to manipulate images in multiple steps. As an outcome, their model produces better manipulation results than previous methods while preserving image areas that are irrelevant to the manipulation prompt. In order to improve the disentanglement of different attributes and the limited efficiency of ManiGAN [51], Li et al. [52] suggest a lightweight network containing a novel word level discriminator and significantly reducing the number of parameters in the generator network. The input to the generator consists of text data, which is encoded by a bidirectional RNN [104], and the image features, which are encoded once by VGG-16 [87] and once by Inception-v3 [93]. Text and Inception-v3 image features are concatenated and fed into a series of upsampling, residual blocks and a final generator network producing the manipulated image. In an intermediate stage the image-text affine combination module from [51] is used twice to fuse text and

image features by using the INception-v3 and the VGG-16 image features consecutively. The discriminator produces word level training feedback for every adjective and noun in the manipulation prompt based on a cosine similarity between the embedded text input and the manipulated image. Both methods by Li et al. [51, 52] produce visually appealing results, especially on the less complex CUB bird dataset [99].
Recently, the image manipulation task has received increasing attention, gradually crossing the boundaries between image generation and image manipulation. Patashnik et al. [71] and Gal et al. [30] use large generative models for image generation and control the generation process by interpreting the latent space. Even more recently, diffusion models [65] are getting a lot of attention for this task. Denoising diffusion probabilistic models work under the objective of generating an image by gradually denoising a sampled random noise input. These methods make it possible to generate or manipulate high-resolution, highly realistic images on complex data sets, but have the disadvantage of reacquiring comparatively large amounts of computing resources.

2.4 Data Augmentation

In general, the goal of data augmentation is to increase the amount and the diversity of the training data without explicitly collecting new data. Common techniques augment by adding modified copies of the original data or synthetic data. This procedure is used to achieve better model performance and generalization capability. These improved results can be realized because the enriched data acts as a regularization and reduces overfitting of the model by extracting more information from the initial training data while preserving the embodied information embodied [85]. While data augmentation is almost a standard technique when training vision models, it is still comparatively underexplored in NLP [29]. This may be due to the discrete nature of language, which makes augmentation challenging. Compared to that, vision allows for simple transformations which can provide additional training input while preserving the label information.

2.4.1 Image Augmentation

Basic image augmentation techniques can already be found in training pipelines of early deep learning approaches like *LeNet-5* [50] or *AlexNet* [48]. Krizhevsky et al. [48] use random cropping to increase the number of data samples in combination with horizontal flipping and colour augmentation. More recent advanced techniques combine these basic image manipulations in automated augmentation approaches, in which the augmentation hyper parameters are learned during training [17]. Other augmentation methods work on a regional level of the image, *e. g. CUTOUT* [24] augments through a regional dropout removing random regions of the input image. Zhang et al. [110] propose *MIXUP* which extends the training data by incorporating linear interpolations of feature vectors and labels. Similarly, *CUTMIX* [108] replaces a region of a random image by a patch of a second random image and linearly interpolates the labels accordingly in proportion to the sub-region areas. Dvornik et al. [27] propose a COPY-PASTE augmentation for object detection. It is realized by leveraging segmentation annotations combined with a context model. This model estimates the likelihood of a particular object category being present in a certain part of the image given its neighborhood. Following this idea, Ghiasi et al. [31] show that the simple strategy of randomly picking objects and pasting them at random locations of the target image provides significant improvement for the task of instance segmentation.
Rather elaborate augmentation approaches base on the success of GANs [32]. Hereby GANs are used to

accumulate new training data through image synthesis [106] or style transfer [86]. This approaches come with a high computational overhead in comparison to the previously presented methods. However, this is acceptable in areas like medical imaging [106], where data curation is highly complex and expensive.

2.4.2 Text Augmentation

A widely used primitive approach to apply data augmentation to textual data are rule based techniques. Wei and Zou [103] propose easy data augmentation using several random token level perturbations such as random insertion, deletion and swap resulting in increased performance on various language classification tasks. Introduced by Zhang et al. [112], synonymous replacing augmentation translates the natural idea of augmenting by replacing words in a text with their synonyms. Given a synonym thesaurus for a number of words, each word is given a list of synonyms sorted by the semantic closeness to the most frequently seen meaning of the respective word. In order to generate a new sentence that matches an existing sentence consisting of n words, the m words are determined which are also present in the dictionary. The number r of words to be replaced is randomly chosen and the probability of number r is determined by a geometric distribution with parameter p in $P[r] \sim p^r$. The index s of the selected synonym likewise underlies the probability distribution $P[s] \sim q^s$. Hence, the further away a synonym is considered to be from the word's most frequently seen meaning, the lower the probability that this synonym is selected. Şahin and Steedman [80] propose dependency three morphing which simply swaps or deletes children of the same parent using dependency annotated sentences. Another technique when augmenting natural language is inspired by the *MIXUP* [110] and *CUTMIX* [108] methods from the vision community. When applying this idea to textual data, Chen et al. [14] carry out the interpolation on the embeddings or hidden layer representation. Another category are model-based techniques using sequence-to-sequence and language models for augmentation. Sennrich et al. [83] propose *Backtranslation*, here, a sequence is translated into another language and then back into the original language to obtain an augmented version of the original sequence. Other approaches [49, 105, 46] train models learning to generate diverse paraphrases. Ng et al. [64] propose a corrupt-and-reconstruct approach by masking arbitrary words of the sequence and reconstructing it using *BERT* [22].

2.4.3 Augmentation for Image Captioning

In general, there has been little research in the area of augmentation for multimodal data and especially for image-text data. Moreover, existing work tends to augment only a single modality.
Previous work [100, 13, 42] explores basic image augmentation techniques when training captioning models, as described in Sec. 2.4.1. Wang et al. [100] use cropping, scaling and flipping of the image modality. Bujimalla et al. [13] focus on improving robustness of captioning models by training on motion blurred samples. Katiyar and Borgohain [42] use random horizontal flipping and random perspective transformations when training a image captioning model. Takahashi et al. [94] augment both modalities for image-caption retrieval by composing new images out of four random patches of random images and averaging the four encoded captions to retrieve the augmented caption.
Other approaches apply augmentation to the text modality instead of manipulating the visual part of the data. For example, Cui et al. [18] use trivial text augmentations to generate pathological negative examples when training a learning-based metric for image captioning. Random captions are sampled from other images in the training set. Furthermore, a number of words in the caption is randomly permuted and words are replaced by random words of the vocabulary. Klein et al. [45] introduce an attribute insertion method to extend ground-truth captions with diverse style attributes. This is realized by inserting a sampled adjective from an

additional attribute dataset into the caption with the help of a part-of-speech (POS) information. Li et al. [53] propose the *Extract-Retrieve-Generate* data augmentation framework, extracting style phrases from small-scale stylized sentences and drafting them to large-scale factual captions. Hereby, a multimodal scene retriever is utilized to access a similar stylized image caption pair for each factual image caption pair. It generates a stylized caption by merging the factual caption and the style phrase of the queried pair. Following a similar approach, Mahajan and Roth [59] obtain pseudo captions that are used during training to supplement the annotated captions as described in Sec. 2.2. Atliha and Šešok [7] use *Contextualized Word Embeddings* to augment image captions, similar to the augmentation technique used in [46] for text-classification. Given a caption consisting of n words and a fixed language model that can predict the probability of a certain word appearing in a particular context. Analogous to the synonymous replacing augmentation [112] presented in Sec. 2.4.2, a geometric distribution is used to determine the number r of the n words to be augmented. For each of the r words the entire caption, except for the word to be augmented, serves as the context for the language model. Subsequently, the word to be augmented is replaced by the most probable predicted word. This procedure is repeated iteratively for all words of the caption. The language model used for this task is *BERT* and was already introduced in Sec. 2.1.5. Modest improvement is achieved over several metrics with several models. Moreover, it is shown that the word level augmentation saturates at some point in enriching the data since each word, in a given context, has only a limited number of words it can be replaced by.
Analyzing the approaches described above, it can be concluded that a lot of research insights have been achieved regarding data augmentation on a single modality. Despite this, there is a lack of methods augmenting multimodal data beyond augmenting a single modality in a way to still match the second modality. Therefore, it can be assumed that the full potential of the two paired modalities is not fully accessed.

2.4.4 Word Vector Representations

In order to be able to process textual data it is common practice to represent each word of the used vocabulary by a vector. Using basic one-hot encoding comes with the disadvantage of very large vectors when using increased vocabulary sizes. Furthermore these contain no additional information besides the direct word encoding. Based on such embeddings, it is not possible to determine word similarity since all vectors have the same distance to each other. This led to the development of dense representations that reflect not only the actual word, but also its context such that semantic similarity between words can be retrieved. Two popular approaches are Word2Vec [61] and GloVe [72].
Word2Vec trains a feed-forward neural network to subsequently obtain the hidden layer values as final word embeddings. The learning process is based on the context words surrounding the word to be embedded in the test data used for training. Two approaches are proposed. The skip-gram model predicts the context words on the basis of the original word. Vice versa, the continuous bag of words model predicts the original word based on the context words. GloVe combines this local context window method with a global matrix factorization technique. Hereby learning is performed on aggregated global word-word ratios of co-occurrence from the training corpus. This results in the low dimensional word representation. Both approaches produce high quality dense vector representations used in a variety of NLP tasks.

2.4.5 Image-Text Similarity

When augmenting the paired data it has to be ensured that the augmented text and image are still aligned. This can be measure by a image-text similarity. Several proposed approaches to conquer this issue are based on the concept of mapping text and image into a common Euclidean space in which the geometry is representing

meaningful semantic relations.
Xu et al. [104] propose a *Deep Attentional Multimodal Similarity Model* (DAMSM) which is used as an additional loss term for text-to-image generation. The model learns a bi-directional LSTM [82] as a text encoder and a Inception-v3 model to encode the image [93]. To quantify the alignment of an image-sentence pair, an attention model between image and text representation is used. The attention model is built upon the cosine product similarity between every word of the sentence and every image sub-region. Moreover, the model is trained in a semi-supervised manner where the only supervision mechanisms are the matched image-caption pairs. [51] leverage the attention-driven image-text matching score to quantify text-guided image manipulation results. The proposed manipulation precision metric aims to simultaneously measure the quality of generation and reconstruction by combining L_1 pixel difference between an image and its manipulated version with the attention-driven image-text matching score of the text used to guide the manipulation. Similarly to [104], Li et al. [52] use the DAMSM as apart of the training loss when proposing a lightweight GAN for text-guided image manipulation.
Radford et al. [74] propose Contrastive Language-Image Pre-training (CLIP) which is trained on predicting image-text pairs for 400 million examples collected from all over the internet. In the context of [75], CLIP is used to re-rank text-to-image generation results based on image-text similarity. Similar to the DAMSM method described above, the CLIP architecture mainly consists of an image and a text encoder. Several different architectures were presented, which differ in terms of the used image encoder. The text encoding of all proposed architectures is done by a slightly modified Transformer [95] while different ResNet [35] and Vision Transformer [26] configurations are used for image encoding. The underlying pre-training objective is fairly trivial as the model only needs to predict which text as a whole is paired with which image. Given a batch of N image-text pairs, the training objective of the model is to predict which of the $N \times N$ possible pairings actually occurred. Thereby, a multimodal embedding space is learned by jointly training both encoders. The objective is to maximize the cosine similarity of real image-text pairs while minimizing the similarity for incorrect pairs which is done by optimising a symmetric cross entropy loss over the similarities. The pre-trained model combined with natural language prompting enables a zero-shot transfer to various downstream tasks at an on par performance with task specific supervised models.

3 Approach

In the following, a new method for multimodal data augmentation in the context of image captioning is proposed. The developed CutSwap method allows a lightweight and controllable augmentation of multimodal image-text data of both modalities. Hereby, the underlying idea is to swap specific image areas showing certain objects with a contentwise reasonable image patch from a different image. Correspondingly object and attribute tokens in the caption are replaced by those describing the content of the used image patch. This is covered in detail in Sec. 3.2. The initial idea of this research was to use text-guided image manipulation GANs in order to augment the multimodal data. First experiments showed that modern lightweight GAN architectures are not yet able to perform satisfactory image manipulations on complex visual data. This is described in more detail in Sec. 3.1.

3.1 Exploring GAN-Based Image Manipulation for Multimodal Data Augmentation

Initial inspiration for this research is the work of Li et al. [52] in the area of text-guided image manipulation using GANs, presented in Sec. 2.3. Compared to state-of-the-art methods in the field of image synthesis [41], image manipulation allows for similarly realistic if only partially new images at a fraction of the computational effort compared to image generation approaches. [52] is an improved version of ManiGAN [51] containing less than $50\,\%$ of its parameters while yielding better performance on the CUB [99] and COCO [56] dataset. In addition, this lightweight generative method offers an ideal interface when working with image-text captioning datasets due to the text-guidance. The initial idea is to modify the caption via text augmentation in such a way that visually expressive words are replaced by other visually expressive ones. Based on this, the lightweight GAN is leveraged to manipulate the visual modality to align with the previously augmented caption. In order to introduce new information into the dataset via such augmentation, it is crucial to have extensive manipulation capabilities. Furthermore, a reliable manipulation is desirable to avoid unaligned image-text pairs in the augmented data.

In a first step, the results of Li et al. [52] are reproduced and the image manipulation capability of the model is explored qualitatively. This is done for the challenging COCO dataset. Figure 3.1 shows a selection of good manipulation examples. Here, the desired changes are made to the image section described in the text description without effecting the rest of the image. These qualitative examples are highly similar to the examples shown in [52] what is remarkable considering how lightweight the network is.
Unfortunately, it requires carefully constructed manipulation descriptions to achieve such results. In general, one is more likely to obtain images like the examples shown in Figure 3.2. Here, the manipulation prompt is not applied to the image, but instead the colors of the entire image are shifted. Furthermore, it can be observed that image artifacts occur as shown in the example on the very right of Figure 3.2. This is a results of the image decoding process from the feature representation. In addition, bubble-like artifacts can be observed, which are likely due to the attention applied to the input of the image-text affine combination module. Another

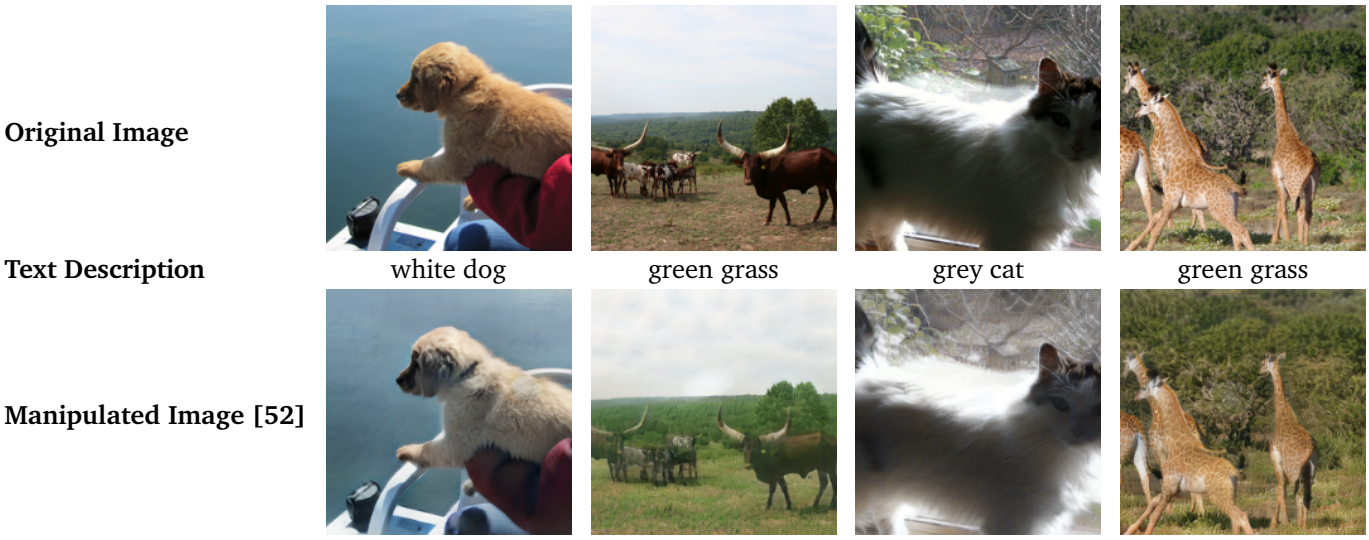


Figure 3.1: Selected qualitative manipulation examples using text-guided lightweight GAN on the COCO dataset.

observation is that the method produces better manipulation results when using short, precise manipulation prompts. This may be due to the fact that all words that can not be defined as nouns or adjectives are masked for the training loss calculation.

[52] uses the Fréchet inception distance (FID) [36] score for evaluation which is explained in Sec. 4.3. To understand how these fairly limited manipulation results are related to the comparatively good FID score, the evaluation is explored in more detail. It should be noted that there is no standard evaluation protocol for the task of image manipulation since manipulation has been hardly researched so far and quantification, as with generative tasks in general, is challenging. Recent methods [71, 30] in the field of image manipulation

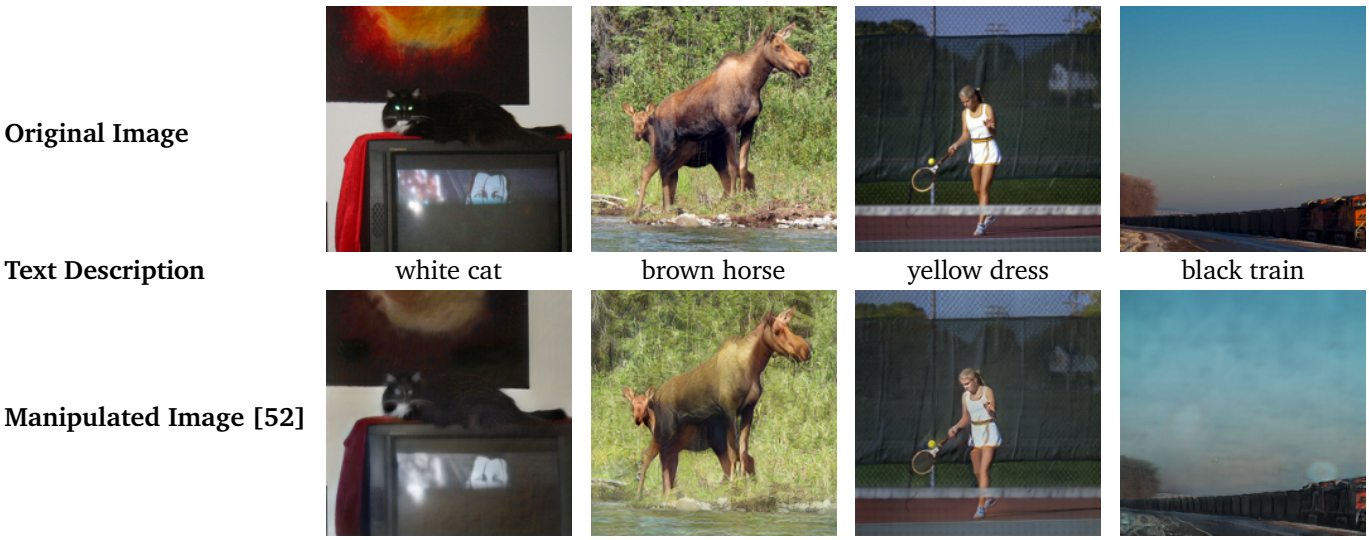


Figure 3.2: Random qualitative manipulation examples using text-guided lightweight GAN on the COCO dataset.

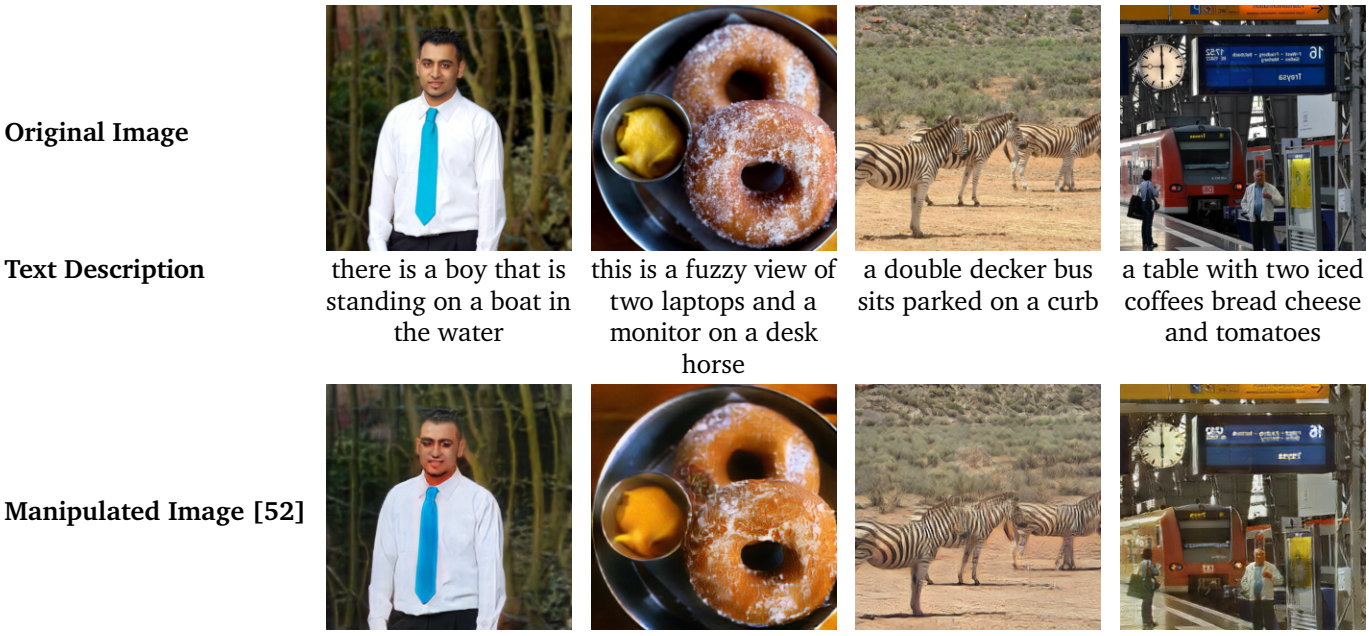


Figure 3.3: Random qualitative manipulation examples following the evaluation protocol of text-guided lightweight GAN on the COCO dataset.

therefore focus on qualitative visual examples as well as comparisons for evaluation.
In order to evaluate their method on the FID, Li et al. [52] manipulate each image of the test dataset and compare them to the original data. To do so, randomly chosen input images are edited by randomly selected text descriptions.

On the CUB bird dataset [99], which is mainly used in [52], this approach is reasonable. CUB consists of a variety of images showing different bird species in their natural habitat. In simple terms, the images show differently colored birds on a mostly blurred, monotone and natural background. This leads to image captions focusing on the visual properties of the birds. Accordingly, by swapping the captions, one receives quite useful manipulation prompts. Unfortunately, this is not the case for the COCO dataset as shown in Figure 3.3. The COCO dataset describes very distinct scenes so that it is not possible to align an image with a randomly selected caption by performing basic image manipulations. In order to use such a method for evaluation, one could have determined the nearest neighbors of the captions by *e. g.* [28] to omit prompts to which the image can not be aligned with through manipulation. Analyzing the visual examples, it becomes clear that the text descriptions do not match the images at all which finally leads to weak manipulations. Thus, with the exception of artifacts, the manipulated image is very similar to the original image what ultimately results in good FID values.

Summing up, [52] offers insufficient manipulation capacity to generate new information through augmentation. Furthermore, the method is unreliable. However, this is also due to the complex and challenging COCO dataset. When dealing with simpler datasets, the method delivers impressive results, especially with regard to computational efficiency. Similar results where obtained in first qualitative experiments using the larger ManiGAN [51]. Since a goal of this work is to develop a applicable augmentation method, heavier models and approaches such as text-to-image generation are not sensible. Furthermore, neither the required resources are available in many contexts, nor is it reasonable to spend far more computational effort on data augmentation than on the actual task.

3.2 Proposing CutSwap Multimodal Augmentation

In the following, the CutSwap multimodal data augmentation method is introduced. CutSwap is a highly lightweight and flexible approach for image-text data augmentation. Since image captioning pipelines and the multimodal datasets already provide a large amount of information, this approach aims to take an advantage of these existing information instead of introducing additionally computational overhead. By leveraging the entirety of the training data information, basic image and text processing techniques and a multimodal pre-trained method, CutSwap efficiently generates novel meaningful data samples.
This approach is inspired by the work of Mahajan and Roth [59] who show that a single caption can properly describe a number of different images by simply adjusting the object token in the textual description to the object shown in the image. In addition, many modern image captioning methods use image features from object detection networks as the input visual representation. This leads to the hypothesis that realism within the bounding boxes of the individual image features is sufficient and global realism can possibly be neglected. A natural assumption is that the majority of images consists of sections showing one or more objects and sections showing a certain context or background. Correspondingly the textual descriptions consisting of a description of the objects and a description of the context. Combining these findings and assumptions, one arrives at the foundations this method is rooted on.

Given an image composed of image regions showing the object i^o along with image areas showing the context i^m. Correspondingly given a textual description consisting of the object words x^o along with attributes describing it x^a as well as the remaining words of the caption x^m. The latter refers to the caption excluding the object tokens as well as the attribute tokens describing the objects. Thereby, x^m may be a description of the background or the spatial relationships between objects. CutSwap aims to generate meaningful new data by swapping x^o and x^a with a reasonable other object-attribute pair as well as by swapping i^o with a patch showing this respective new object-attribute pair.
The functioning of CutSwap augmentation is schematically visualized in Figure 3.4. First, the augmentation dataset is pre-processed. This step generates knowledge of the available object classes and, hence, object-attribute pairs occurring in the dataset and from which sample they can be retrieved. Furthermore, the object classes are clustered into groups. This aims at obtaining classes that are interchangeable as they appear in a similar context, resulting in meaningful images and captions. In addition, pre-computed object detection bounding boxes are used. These are already available serving as the input visual representation for the captioning model. Given an original image-caption pair of the training data, the text augmentation module swaps x^o and the associated x^a of the image description by a randomly selected object-attribute pair based on the pre-processed augmentation data. Next, the image manipulation module replaces i^o by a randomly selected image patch showing the augmented object-attribute pair. This is performed several times leading to n_i augmented images per augmented caption. Likewise, the whole process is repeated multiple times resulting in n_p augmented captions and $n_p \cdot n_i$ augmented images. To obtain the final augmented image-caption pairs the image-text similarity is determined by using CLIP which is presented in Sec. 2.4.5. Based on this, filtering and re-ranking is carried out.

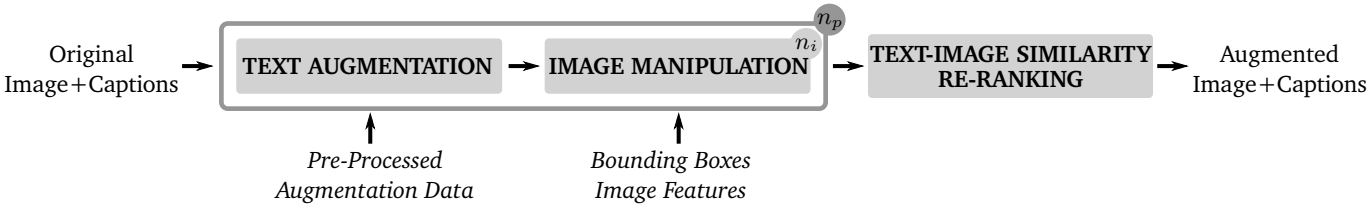


Figure 3.4: Schematic architecture of the proposed CutSwap multimodal augmentation method.

Figure 3.5 shows an augmentation example. The original object $x^o_{orig} = \textit{pizza}$ with the attribute $x^a_{orig} = \textit{plain}$ is replaced by the object-attribute pair $x^o_{aug} = \textit{sandwich}$ and $x^a_{aug} = \textit{double}$ in the caption. Accordingly, the image patch showing the original object i^o_{orig} is replaced by a patch showing the augmentation object i^o_{aug}. In the following, CutSwap is presented in detail based on this example.

Original Image-Caption Pair

A plain pizza is shown on a wooden table with empty plates.

Augmented Image-Caption Pair

A double sandwich is shown on a wooden table with empty plates.

Figure 3.5: Original and CutSwap augmented image-caption pair.

3.2.1 Dataset Pre-Processing

The dataset pre-processing step aims to retrieve condensed dataset information of the data used for augmentation. This process can be divided into two stages. Within the first stage all data is collected, before it is then filtered and processed within stage two.
In the case of the multimodal COCO data, one image i and five annotated image descriptions x are given per sample. The following steps are performed for each image and correspondingly for each image caption. Following Mahajan and Roth [59], tokens corresponding to Visual Genome [47] object classes are separated denoted as object x^o. By using Part-of-speech (POS) tagging [10] and Dependency Parsing [38] those tokens dependent on x^o and POS-tagged as adjectives are separated denoted as attributes x^a. After this step one knows all object classes that are present in the dataset as well as all corresponding object-attribute pairs and the sample visualizing the respective combination. In the second pre-processing stage, the attributes occurring less then five times are filtered out. This is justified for the used COCO dataset by the assumption that there are five captions per image. Therefore, an attribute that only appears in one image but is nevertheless important should be mentioned in all five captions. Additionally, the object tokens are clustered based on their word embeddings. This follows the interpretation of Mikolov et al. [61] stating that word embeddings group similar words, as the vectors are obtained by learning the textual context the embedded word is surrounded by. Accordingly, clustering these vectors should lead to groups of words that can be interchanged in a meaningful way. CLIP word embeddings are used and grouped utilizing Agglomerative Clustering [89]. Experiments leading to this choice as well as more detailed insights are presented in Sec. 4.4.1. For the presented augmentation example the object cluster as well as the corresponding attributes are shown in Figure 3.6.

Object Cluster containing *pizza:*
cake, pizza, sandwich, cupcake, pancake, shortcake, cheese
Attributes for *sandwich:*
italian, faced, cheesy, fresh, small, delicious, full, messy, tasty, meaty, next, hearty, thick, double, deli, long, loaded, nice, large, hot, toasted, huge, big, looking, open, healthy, giant, half

Figure 3.6: Clustered object classes and attributes for CutSwap augmentation example.

3.2.2 Text Augmentation

Given an original caption x_{orig} for augmentation, it is split into x^o_{orig} for the tokens being visual genome classes. If the caption contains more than one object class, only one of the objects will be randomly selected. Similar to the pre-processing, dependency parsing and POS tagging is used to obtain the attribute words x^a_{orig} of x^o_{orig}. A token is denoted as an attribute in case it is dependent on x^o_{orig} as well as POS tagged as an adjective. Next, an augmentation object x^o_{aug} is determined by using the pre-processed augmentation data. Hereby, the closest object cluster is found by calculating the cosine similarity between the CLIP embedding of x^o_{orig} and all object cluster centers. Next, x^o_{orig} is replaced by a randomly selected object class token x^o_{aug} of the closest cluster. Similarly, one of the attributes occurring with x^o_{aug} is randomly selected. Consecutively, x^a_{orig} is replaced by x^a_{aug}. Furthermore, some frequent exceptions are handled. In the case that the original caption contains multiple consecutive attribute tokens, these are all replaced by one single augmentation token. For example an original caption containing two attributes connected by *and*, all three tokens are replaced by just one x^a_{aug} token. Moreover, a distinction is made between plural and singular object tokens. Plural x^o_{orig} are only swapped with plural x^o_{aug} and vice versa. Furthermore, an augmentation is only valid when either x^o_{orig} or x^a_{orig} has changed with respect to the original caption. Qualitative evaluation showed that there exist only rare cases in which CutSwap produces corrupt augmentations. Those are mainly due to errors in POS tagging, dependency parsing or low quality original captions. Qualitative experiments show that this setup leads to syntactically and semantically accurate augmentations for the vast majority of captions. This can be observed in Figure A1 and Figure A2 showing random qualitative CutSwap augmentation examples and selected failure cases. The architecture of the text augmentation module is schematically visualized in Fig. 3.7.

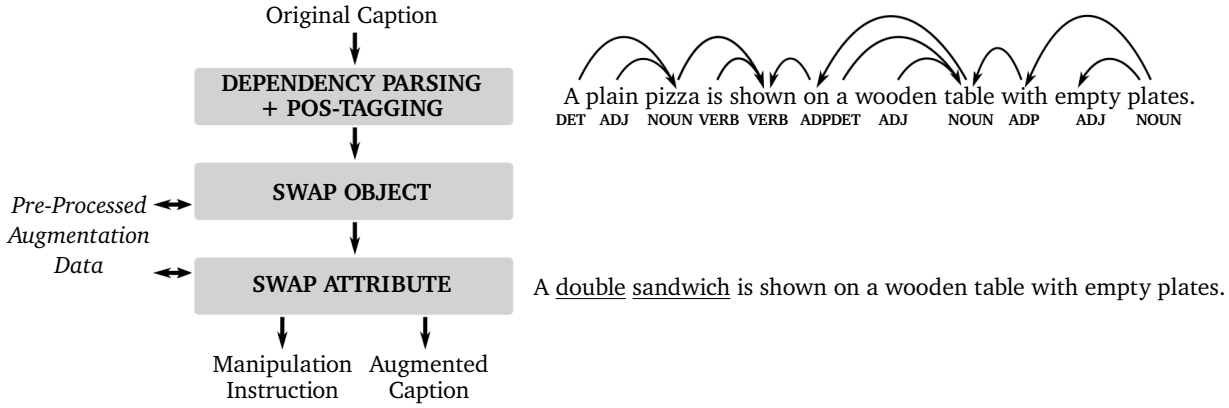


Figure 3.7: Schematic architecture of the text augmentation module in CutSwap.

3.2.3 Image Manipulation

Based on the previously conducted text augmentation, the object-attribute pair $\{x^o_{aug}, x^a_{aug}\}$ used for augmentation is given. Additionally, the original image as well as the bounding boxes and predicted object classes of the image features are set. In a first step, the bounding boxes for the x^o_{orig} class are determined in the original image and NMS is applied. Despite the fact that [4] already uses NMS when retrieving the image features, it was found that repeating NMS with a higher intersection over union threshold is beneficial for augmentation to avoid multiple boxes for the same object. This results in the patches of the original image i^o_{orig} that CutSwap aims to replace. Furthermore, augmentation is only applied when the i^o_{orig} area, compared to the area of the original image, is neither too small nor too big. This results from the intuition that neither swapping a tiny fraction nor replacing the whole image can lead to a meaningful sample. The next step operates based on the manipulation instruction derived from the pre-processed augmentation data by the text augmentation module. Given the set of images for augmentation i_{aug}, all potential image patches are retrieved by respectively applying NMS to the bounding boxes of the class corresponding to x^o_{aug}. Figure 3.8 shows all image feature bounding boxes after NMS per object class for the original image used in the example as well as the source image used for augmentation.

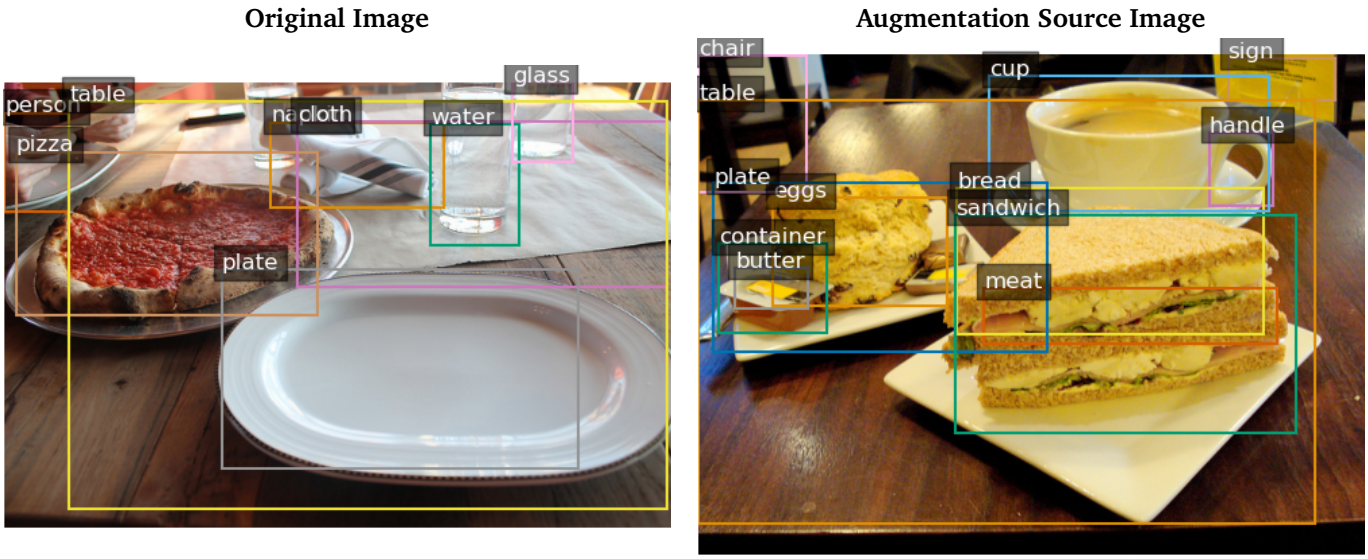


Figure 3.8: Original and augmentation source image showing all image feature bounding boxes after NMS.

On top of the potential patches for augmentation, filtering is applied to ensure that the pixel density as well as aspect ratio is similar to the original image patch which is about to be replaced. Thereafter, n_i patches are randomly selected from the remaining set. The augmented images $\sum_{n=0}^{n_i} i_{aug_n}$ result from resizing the source image patch i^o_{aug} and replace i^o_{orig}. In the case of more than one i^o_{orig} patches, all are replaced by the same i^o_{aug}. Furthermore, the edges between the original image context i^m_{orig} and i^o_{aug} are blended over. In addition, it is ensured that image areas of other object classes mentioned in the caption are not covered by i^o_{aug}. This is realized by omitting them in i^o_{aug} if the overlap is small. Most hyperparameters can be chosen by logical assumptions so that only rare extreme cases are avoided. Since this generative augmentation is difficult to quantify and a hyperparameter study envolving the whole training pipeline is not feasible, the parameters are chosen based on qualitative results. CutSwap replaces at least $10\,\%$ and at most $70\,\%$ of the original image. Furthermore not more than $50\,\%$ of the i^o_{aug} is omitted. Figure 3.9 visualizes the schematic process of the image manipulation module.

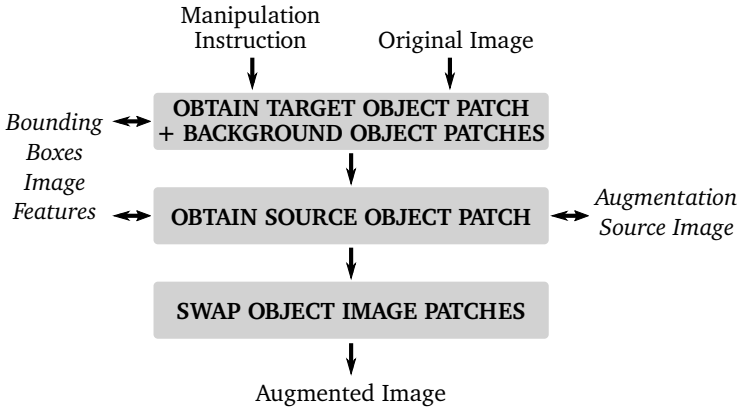


Figure 3.9: Schematic architecture of the image manipulation module in CutSwap.

3.2.4 Similarity Re-Ranking

As both text augmentation and image manipulation operate without any supervision while partially relying on model predictions, errors are inevitably introduced. Poor image feature bounding box predictions, faulty POS tagging or dependency parsing as well as poor quality of original captions result in likewise poor augmented examples. Inspired by the re-ranking approach used in [75], the multimodal pre-trained CLIP model is implemented to threshold and re-rank the augmented image-caption pairs in a final CutSwap step. In order to do so, the augmented image and the corresponding augmented caption are encoded by CLIP and the cosine similarity is calculated for every pair. Augmented pairs yielding a similarity beneath a certain threshold are discarded while all others are ranked by their similarity scores. For the augmentation example shown, augmentations and the respective similarity scores are shown in Figure 3.10.

Caption	A x^a_{aug} x^o_{aug} shown on a wooden table with empty plates.				
Image					
x^a_{aug}	lit	small	tall	thick	double
x^o_{aug}	cake	sandwich	cake	pizza	sandwich
Similarity	0.276	0.312	0.322	0.335	0.351

Figure 3.10: CutSwap similarity re-ranking example showing augmented caption, augmented image and cosine similarity of CLIP embeddings.

3.2.5 Comparison to Existing Methods

Analyzing CutSwap, a certain degree of similarity to existing augmentation approaches cannot be denied. However, to the best of my knowledge, CutSwap is the only method augmenting multimodal data of both modalities resulting in meaningful new data samples.
In general, one could observe parallels to the methods RICAP [94], CUTMIX [108] and COPY-PASTE [27, 31]

which are presented in Sec. 2.4. RICAP is applied to the closely related task of image-caption retrieval, also augmenting both modalities. Hereby, four random patches of four random images are merged into one. The corresponding four image descriptions are encoded and a weighted mean based on the respective areas is applied. This results in unnatural and for humans non-interpretable images and text descriptions. CUTMIX operates similarly by replacing a part of an image by a random patch of a second image. The label is obtained by forming the mean of the labels weighted by the area share of the respective images. Likewise, this results in non-interpretable augmentations. Parallels also exist with regard to the COPY-PASTE augmentations for object detection [27] and instance segmentation [31]. Both methods insert parts of other images into the augmented image and adjust the annotation accordingly. Dvornik et al. [27] realize this by using a network to predict the position and the object class to be inserted based on the neighbourhood of an image region. Whereas with CutSwap this is obsolete with regard to the visual augmentation since the area to be replaced is known. This is because image sections showing an object are replaced instead of adding additional objects into the image. However, CutSwap also determines the possible object classes to be used based on a context. This context is provided by the word embeddings clustering of the object classes. A similar core idea to the one of CutSwap was recently brought up by Feng et al. [29]. Their idea consists of combining *CUTMIX* and caption editing to achieve multimodal augmentation. Even more recently, Hartmann et al. [34] mention a similar idea for an application in interactive learning for image captioning to multiply human feedback through augmentation. Beyond the loose formulation of the idea, neither methods nor results are presented by either.

4 Experiments

This chapter aims to present the most important experiments that were conducted in the course of this work. At the beginning, implementation details, datasets and metrics are presented. Subsequently, individual elements of the CutSwap method are examined in more detail. This is followed by the experiments on the augmented data. Finally, the training setup, results and the respective analysis is provided.
CutSwap augmentation is examined for the task of image captioning using the deterministic UpDown model by Anderson et al. [4]. Furthermore, CutSwap is examined for the task of diverse image captioning using the stochastic COS-CVAE method Mahajan and Roth [59]. Both models are introduced in Sec. 2. The overall goal is to investigate whether the presented CutSwap method leads to better captioning models by enriching the training data through augmentation. Moreover, no additional data is introduced. This is investigated in a standard augmentation setup. The method augments the training dataset with additional samples while the training setup remains identical to the baseline. This challenging setup allows to directly trace back performance differences to the method as no other modification are made. Furthermore, a lightweight setup that is consistent with the method is used. Additionally, it should be noted that it is difficult to directly quantify the augmentation task due to its generative characteristic. Therefore, many experiments are performed qualitatively. The overall goal is to use augmentation to improve the performance of the captioning models so that the method can be indirectly quantified.

4.1 Implementation

CutSwap runs primarily on the CPU and is implemented mainly in NumPy [33]. Furthermore, SpaCy [38] and NLTK [10] are used for language processing and pillow [16] for image processing. For re-ranking, the official PyTorch [70] implementation of CLIP [74] is adopted. Image features of the augmented images are retrieved using the PyTorch implementation [107] of [4]. For the UpDown image captioning model the official pytorch re-implementation from [1] is used, whereby the entire setup is adopted. Similarly, the official implementation of the COS-CVAE [59] diverse captioning model is adopted and both the training and evaluation setup are followed. All experiments were performed on a NVIDIA GTX 2080 (11GB) GPU.

4.2 Datasets

This work is mainly based on the COCO dataset [56] which is a commonly used, highly diverse and challenging dataset in the field of image captioning. In order to examine the model performance in more detail, the Nocaps [1] dataset is used in addition. Moreover, this allows to access generalization capabilities. Nocaps contains new images and objects and divides these into three groups according to their semantic distance to COCO.

4.2.1 COCO

Microsoft COCO is one of the most frequently used datasets. Apart from the image captioning task it is also used for object detection and image segmentation. The COCO dataset captures complex scenes of numerous everyday life categories that can be found in a natural environment. The dataset consists of more than $120\,000$ labelled images with object annotations of 80 different categories as well as five different human annotated captions per image. The 2017 split consists of $118\,287$ training images as well as 5000 validation images. There is also an official test set containing $40\,775$ images with 40 confidential captions each. Consistent with Mahajan and Roth [59] and Anderson et al. [4], $118\,287$ train, 4000 validation, and 1000 test images are used.

4.2.2 Nocaps

The Nocaps dataset is a high-quality benchmark consisting of $166\,100$ human-generated captions describing $15\,100$ images. It yields over 400 object classes more than COCO and represents more visual concepts. Moreover, Nocaps contains more object classes as well as object instances per images and does not include images showing only one object class which is however the case for $20\,\%$ of the COCO images. The benchmark was very carefully designed regarding the distribution of the occurring classes over the dataset and the quality of the human annotated captions. The data is split into a validation set containing 4500 images and a test set containing $10\,600$ images. Additionally, those sets are split into three subsets depending on the semantic distance of the image to the COCO data. This allows to quantify the generalization capabilities of a model in more detail. *In-domain* images contain only objects of the 80 COCO object classes as well as 39 classes that are not COCO classes but were mentioned more than 1000 times within the COCO captions. *Near-domain* images contain both the in-domain and out-of-domain object classes. *Out-of-domain* images do not contain any of the *in-domain* classes and are visually very distinct from COCO images. Subsequently, the captions are not accessible and evaluation is only possible through an evaluation server. Furthermore, the evaluation protocol, more precisely the evaluation server, accepts only one caption per image. Furthermore the generalization setup and especially the unseen object classes make it difficult to use consensus re-ranking [60] to reduce the diverse captions to a single one. This is due to the fact that the re-reanking is based on the training data. As a result this leads to a merely qualitative evaluation of diverse image captioning models when using Nocaps.

4.3 Metrics

Image Captioning and Image Manipulation use different metrics for evaluation, these are described in the following section.

4.3.1 Image Captioning

The overall goal of language generation is to obtain machine generated texts that are on par with those authored by humans. Therefore, the overall requirement for metrics in this field is to reflect human captioning quality in the best possible way. First and foremost, the caption should describe the image semantics completely and accurately. Moreover, it is required to be grammatically correct and fluent. To avoid time-consuming and cost-intensive human evaluation and to enable the reproducibility of results, automated methods have been developed to accomplish this evaluation. In the following, the metrics used in this work will be discussed in

more detail. First, standard image captioning metrics will be introduced, which are then followed by methods quantifying the diversity of captions.

Bilingual Evaluation Understudy (BLEU) [68] is a precision-oriented metric which is designed for machine translation and based on *n*-gram precision. Having a generated candidate translation C and multiple reference translations C', then for the example of n=1 (unigram), the *n*-gram precision is calculated by counting the words that appear in the generated as well as in any reference text divided by the total number of grams in the generated text. The modified n-gram precision which is used here, is extended by the assumption that a reference word is considered as exhausted once it has been matched. This results in clipping the total count of each generated word accordingly to Equation 4.1.

$$\begin{gathered} Count_{clip} = min(Count,\ Max\_Ref\_Count) \\ p_n = \frac{\sum\limits_{C \in \{Candidates\}} \sum\limits_{n\text{-}gram \in C} Count_{clip}(n\text{-}gram)}{\sum\limits_{C' \in \{Candidates\}} \sum\limits_{n\text{-}gram' \in C'} Count(n\text{-}gram')} \end{gathered} \tag{4.1}$$

Candidate translations that are longer than their reference get penalized by the modified *n*-gram measure. With the help of a multiplicative *brevity penalty* factor BP, candidates shorter than their reference are penalized as well. First, the geometric mean is taken over the N modified *n*-gram precision scores of the test corpus and positive weights w_n summing to one. Then the result is multiplied by an exponential BP. Equation 4.2 shows BP results from the length of the candidate translation c and the length of the reference r.

$$\begin{gathered} BP = \begin{cases} 1 & \text{if } c > r \\ e^{(1-r/c)} & \text{if } c \leq r \end{cases} \\ BLEU = BP \cdot \exp\left(\sum_{n=1}^{N} w_n \log p_n \right) \end{gathered} \tag{4.2}$$

Consensus-Based Image Description Evaluation (CIDEr) [96] is designed to reflect the human judgment of consensus on image caption quality. A particular feature of this method is that often occurring n-grams are given less influence because these are likely to be less informative. This is realized by a *Term Frequency-Inverse Document Frequency* (TF-IDF) weighting for each *n*-gram. As expressed in Equation 4.3, the score for n-grams of length n results from the average cosine similarity of the candidate sentence c_i and the reference sentence s_{ij} while simultaneously accounting for precision and recall. Here, $g^n(c_i)$ is a vector calculated using TF-IDF and $\|g^n(c_i)\|$ being its magnitude. The same accounts for the reference $g^n(s_{ij})$. The final score equals the sum of the weighted scores for the different *n*-grams. Usually uniform weights $n = 1/N$ and $N = 4$ are used.

$$CIDEr(c_i, S_i) = \sum_{n=1}^{N} w_n \left(\frac{1}{m} \sum_j \frac{g^n(c_i) \cdot g^n(s_{i,j})}{\|g^n(c_i)\| \|g^n(s_{i,j})\|} \right) \tag{4.3}$$

To make the metric even more stable against gaming, a Gaussian penalty factor is added so that candidate sentences should be as close as possible to the length of the reference sentence. In addition, no stemming is performed and the number of a specific *n*-grams of the candidate is clipped to the number of occurrences in the reference. This modified version is called CIDEr-D. However, in the following it is referred to as CIDEr.

Recall-Oriented Understudy for Gisting Evaluation (ROUGE) [55] is a recall-oriented metric designed for text summarization. In the following, focus is put on the ROUGE-L version, which is the most widely used version, especially for captioning. The basic idea of the metric is to use the overlapping subsequence of words between a candidate sentence and its reference to quantify the quality. Moreover the words do not have to appear strictly one after a other. Given two sequences c and s of length m and n, the longest common subsequence (LCS) is the common subsequence with maximum length. Having *Recall* and *Precision* calculated as $R_{lcs} = LCS(c,s)/m$ and $P_{lcs} = LCS(c,s)/n$, the score is computed according to the F-Score as shown in Equation 4.4. Furthermore the hyperparameter β is chosen to 1.2, so recall is slightly favoured by the metric.

$$ROUGE = \frac{(1+\beta^2)\, R_{lcs}\, P_{lcs}}{R_{lcs} + \beta^2\, P_{lcs}} \tag{4.4}$$

Metric for Evaluation of Translation with Explicit Ordering (METEOR) [21, 9] is a machine translation metric based on precision and recall. The score is calculated by generating an alignment between the sentences of candidate c_i and reference r_{ij} while minimizing the number of chunks ch of contiguous and identically ordered tokens. As in Chena et al. [15], the alignment is based on exact token matching, synonyms stemmed tokens and paraphrases. Given this alignment m, the score is the harmonic mean of precision P_m and recall R_m for the best scoring reference penalized by Pen, a factor based on the chunkiness of resolved matches. This is shown in Equation 4.5. Hereby, α, γ and Θ are hyperparameters which were tuned to match with human judgement.

$$\begin{gathered} Pen = \gamma\left(\frac{ch}{m}\right)^{\Theta} \quad F_{mean} = \frac{P_m\, R_m}{\alpha P_m + (1-\alpha) R_m} \\ METEOR = (1 - Pen) F_{mean} \end{gathered} \tag{4.5}$$

Semantic Propositional Image Caption Evaluation (SPICE) [3] is designed specifically for the evaluation of image captioning and mainly assesses the semantic and content of the candidate caption. Candidate c and reference r sentences are represented as sets of tuples extracted from their scene graphs G by the function T. Each of the tuples contains one to three elements, representing objects, attributes and relations, respectively. Next, precision P and recall R is calculated over matching tuples in the sets and the score is obtained by the F1-mean. The simplified calculation is shown in Equation 4.6 .

$$\begin{gathered} P = \frac{|T(G(c)) \otimes T(G(r))|}{|T(G(c))|} \quad R = \frac{|T(G(c)) \otimes T(G(r))|}{|T(G(r))|} \\ SPICE = \frac{2PR}{P+R} \end{gathered} \tag{4.6}$$

When working with deterministic captioning models, it is sufficient to evaluate the single caption prediction with respect to accuracy. For the task of diverse image captioning, a second dimension has to be considered. Besides accuracy, diversity is crucial when dealing with multiple captions. Furthermore, bounding conditions

to evaluate the accuracy are needed having multiple captions per image. Therefore, the COS-CVAE evaluation protocol is followed by using oracle evaluation and consensus re-ranking. Oracle chooses a caption with the highest score for every metric. In line with prior work 20 and 100 samples are considered. For a given test image, consensus re-ranking retrieves the nearest neighbor in the training set to use the corresponding captions for accuracy evaluation.

Diversity Metrics In addition to the standard metrics, further measures are used to evaluate the performance of a captioning method regarding the diversity of the generated captions. This is mainly due to the fact that the standard metrics are based on word similarities and therefore have a drawback in measuring characteristics like novelty and diversity. Adopting the evaluation protocol of COS-CVAE, the following metrics are used to quantify diversity. The evaluation is conducted after the 5 best generated captions for each test image are determined using consensus re-ranking based on the CIDEr scores as presented in [23, 101].
Uniqueness describes the share of distinctive captions in relation to the number of all generated captions.
Novel sentences describes the absolute number of captions that were never seen in the training data.
Mutual Overlap – mBLEU-4 measures the diversity across the individual predicted captions. For each of the K generated captions, the BLEU-4 score is calculated *w. r. t.* to the other $K-1$ captions.
Div-n attempts to quantify the diversity of the model by calculating the ratio of the number of different n-grams and the total number of words per set of diverse captions. In the following, n-gram size one and two are used.
Self-CIDEr [102] assesses diversity by applying latent semantic analysis to a kernel matrix consisting of the pairwise CIDEr scores of all generated captions with respect to each other.

4.3.2 Image Manipulation

To quantitatively evaluate image manipulation, previous work [51, 52] makes use of metrics from the field of image generation. In the following, these metrics are discussed in more detail.

Inception Score [81] (IS) With the help of the Inception model [93], a conditioned label distribution $p(y|x)$ is obtained for each generated image x. Based on the assumption that the generated images x are realistic and contain meaningful objects, this distribution should have a low entropy. In general, the generator network G with input z is expected to generate multiple images so that the marginal probability distribution $\int p(y|x = G(z))dz$ should have a high entropy. Combining these requirements by calculating the Kullback-Leibler divergence between the conditional and marginal probability distributions leads to the inception score, shown in 4.7.

$$IS = \exp\left(\mathbb{E}_x \, \mathbf{KL}(p(y|x)\|p(y))\right) \tag{4.7}$$

Fréchet Inception Distance [36] is supposed to improve over the IS by taking original samples into account instead of evaluating only the statistics of the generated images. The general objective of generative models is to produce data matching the observed data. This allows to use the distance between the probability of observing real world data $p_w(.)$ and the probability of observing generated data $p(.)$ as a performance measure. Similar to the calculation of the IS the *Inception* model is used to obtain vision-relevant features that are assumed to follow a multidimensional Gaussian. The distance between the two Gaussians is measured by the *Fréchet distance* d between the Gaussian with mean and covariance (m, C) obtained from $p(.)$ and the

Gaussian with mean and covariance (m_w, C_w) obtained from $p_w(.)$. Thus the FID-score follows as shown in Equation 4.8.

$$FID = d^2\big((m, C), (m_w, C_w)\big) = \|m - m_w\|_2^2 + \mathrm{Tr}\big(C + C_w - 2(CC_w)^{1/2}\big) \tag{4.8}$$

4.4 Exploring CutSwap Method Components

In this section, two main CutSwap modules are examined in more detail. One of the basic intentions of this augmentation is to replace the objects by meaningful other objects. This requires to identify which objects appear together in a similar context within the set of training and augmentation data. Therefore, object clustering is performed during the data pre-processing phase. An additional vital module of the method is the final re-ranking and thresholding of the augmented image-caption pairs using CLIP. This is performed in an attempt to ensure well aligned augmented text-image pairs. In the following, the development of these modules is outlined and the choice of hyperparameters is defined.

4.4.1 Object Clustering

A foundation of CutSwap are groups of reasonable object classes. Such grouping ensures that objects can be swapped without creating meaningless, unrealistic image-text data. This step builds upon word embeddings on the one hand and document clustering techniques on the other hand. Thereby, document clustering constitutes a text clustering task aiming to group similar documents. In the following, the in Sec. 2.4.4 presented word embeddings Word2vec [61] and GloVe [72] as well as the CLIP text embeddings are explored. Regarding the clustering, the present research follows Steinbach et al. [89] and examines K-Means, Bisecting K-Means as well as agglomerative hierarchical clustering. The following experiments are conducted using the 510 Visual Genome object classes retrieved from the augmentation data in the pre-processing step. More precisely, this is carried out on the 332 lemma of the object classes. In order to obtain these object class clusters, the single object tokens are embedded and the resulting vector presentations are clustered.

Figure 4.1 visualizes the objects classes using the different embeddings in a two-dimensional space reduced through t-SNE dimensionality reduction [58]. Hereby, 300 dimensional Word2vec embeddings, x dimensional GloVe embeddings and 512 dimensional CLIP embeddings are used. The visualization by dimensionality reduction is a highly qualitative evaluation. Nevertheless, tendencies that were identified in qualitative clustering experiments can also be identified here. A general problem occurring for all three embeddings is that word tokens with multiple meanings, such as *'apple'*, are encoded by the same vector. In the examples, *'apple'* can refer to the fruit, but also to the tech company or its products. The CLIP embedding is close to *'computer'*, *'cellphone'* and *'laptop'* (Figure 4.1: lower right), thereby clearly referring to the tech company. In contrast to that, the GloVe embedding refers to the fruit as it is close to the embeddings of *'carrot'* and *'broccoli'* (Figure 4.1: mid right). This is similar to Word2vec (Figure 4.1: mid left). These differences can be traced back to the training data with which the word representations were obtained. While such failures are inevitable with basic word embeddings, they are relatively rare. The differences can be interpreted with the more recent data on which CLIP was trained. Figure 4.1 shows that all approaches embed the words surrounded by meaningful other words. Additionally, groupings of similar words as well as larger distances between words that do not appear in a common context can be observed. This results from Word2vec and GloVe being widely used state-of-the-art embeddings trained on large amounts of text data. Similarly, CLIP is

a state-of-the-art method trained on large amounts of multimodal data and based on good text representations. Therefore, qualitative clustering experiments yield very similar results for all three embeddings. Based on these findings, it is proceeded using the CLIP embeddings. As CLIP is already incorporated in the CutSwap method these word embeddings come without any additional cost.

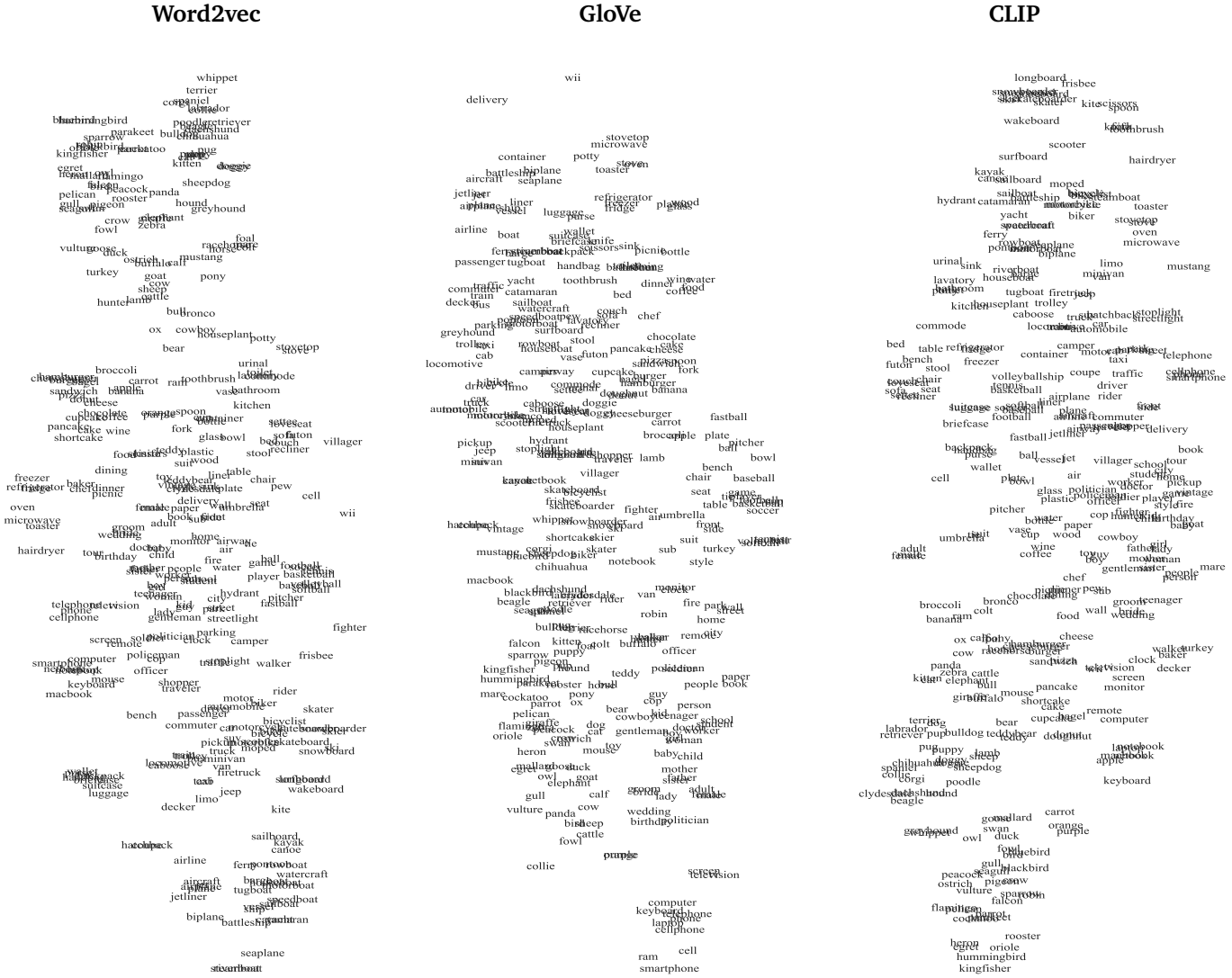


Figure 4.1: t-SNE visualization of Word2vec, GloVe and CLIP word embeddings for visual genome object classes used by CutSwap for data augmentation on COCO.

Based on the CLIP word embeddings, a suitable clustering algorithm is subsequently identified. Following [89], K-Means, Bisecting K-Means as well as agglomerative hierarchical clustering are tested as potential algorithms. The basic K-Means algorithm selects k points as initial centroids. Next, all datapoints are assigned to the closest centroid and the centroids are recomputed for each cluster. The latter step is repeated until the centroids remain constant. Bisecting K-Means combines this procedure with hierarchical clustering. Thereby, K-Means is used with $k = 2$ in the first step. Next the cluster with the highest sum of squared distances is chosen and step one is repeated. This is done until the desired number of clusters is reached. Lastly, agglomerative hierarcical

clustering recursively pairs the clusters with the smallest distance to each other. Initialized by each datapoint being one cluster. Besides, cosine similarity is used which constitutes the common practice when dealing with textual embeddings. Hereby, the euclidean distance is converted into a proportional measure of cosine distance by normalizing the embeddings. Internal quality measures are used to evaluate the performance and cluster quality of the algorithms. Since no external information more precisely ground truth data exist, the sum of squared distances and the silhouette score are used as metrics. Figure 4.2 visualizes these metrics as depending on the number of clusters k. Since the sum of the squared distances is very similar for all clustering algorithms and provides little information about a suitable number of clusters, the focus is on the silhouette score. Based on this, agglomerative hierarchical clustering leads to the best clustering results and the number of clusters is set to $k = 122$.

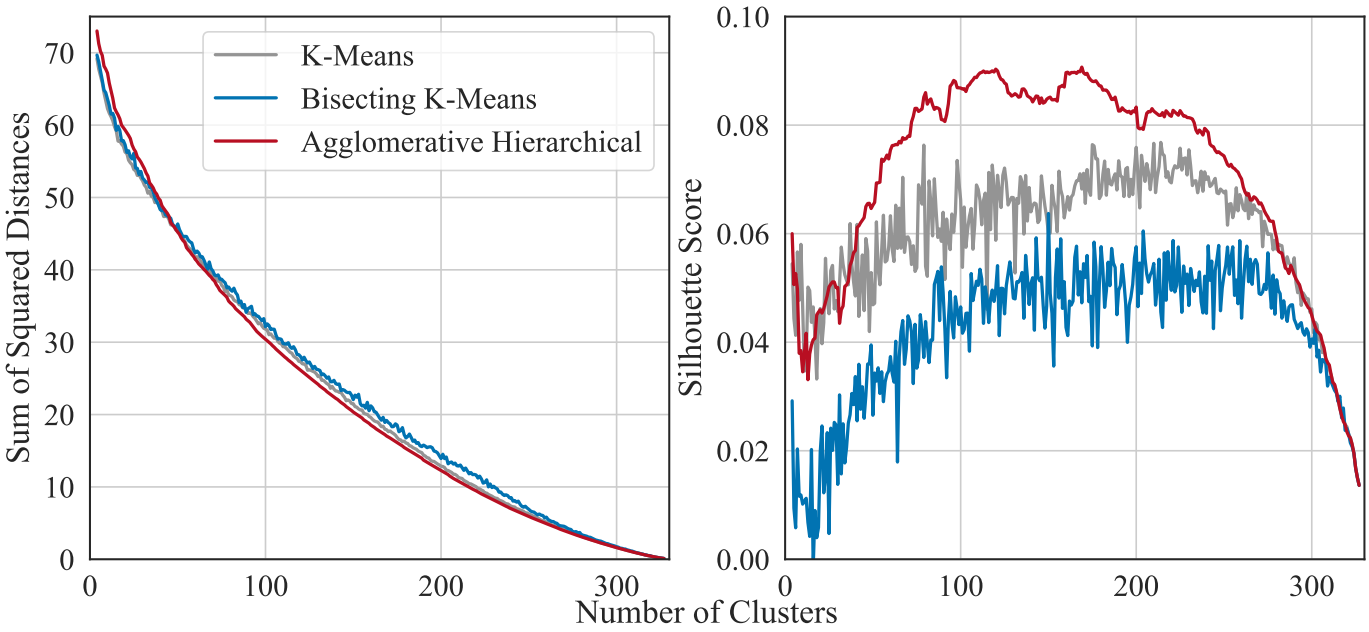


Figure 4.2: Evaluation of the algorithms for object clustering on CLIP word embeddings using sum of squared distances and silhouette score.

This aligns well with the experience gathered while conducting qualitative examples. Furthermore, it results in meaningful grouped object classes. Some random clusters are shown in Table 4.4.1. As the clustering is performed on the lemma of each word. the final cluster consists of all object classes for the respective lemma split by grammatical number. Naturally, some individual words are not grouped in a meaningful way, such as *'umbrella'* in cluster five. However, these are rare exceptions as the majority of groupings consists of words that meaningfully relate to each other.

Table 4.1: Random examples of object class clusters retrieved using agglomerative hierarchical clustering.

Cluster	Singular Object Classes	Plural Object Classes
1	airplane, jet, plane, jetliner	airplanes, jets, planes, jetliners, aircraft, aircrafts, airlines
2	ostrich, pelican, egret, vulture, peacock, heron, flamingo	ostriches, pelicans, vultures, peacocks, flamingos
3	mother, woman, girl, lady, father, sister	mothers, women, womans, girls, ladies, fathers, sisters
4	broccoli, umbrella, banana, chocolate	umbrellas, bananas
5	cup, wine, bottle, coffee, water	cups, bottles
6	car, camper, truck, firetruck, motor, suv, coupe, taxi, hatchback, automobile, cab, jeep	cars, campers, trucks, taxis, automobiles, cabs, jeeps

4.4.2 Augmentation Re-Ranking

Another important part of CutSwap is the re-ranking of the augmented image-caption pairs using CLIP. Hereby, the image-text similarity is determined based on the CLIP embeddings in order to carry out the re-ranking. Furthermore, a image-text similarity threshold is applied. Both text augmentation and image manipulation blindly rely on the annotated data, the pre-processed augmentation data as well as the predicted bounding boxes of the image feature. This results in poorly aligned augmented data in the case of a faulty annotated image caption or an inaccurate bounding box prediction. Aiming to avoid such shortcomings, the CLIP model is leveraged to ensure aligned image-caption pairs.

Following the CLIP training objective, this approach is implemented similar to the proposed CLIP zero-shot classification. Moreover, the idea of using CLIP to re-rank on a multimodal basis is inspired by its usage in the DALL-E [75] text-to-image generation method. Given the $n_p \cdot n_i$ CutSwap augmented image-caption pairs, the feature embeddings of all images and all corresponding captions are computed in a first step. More specifically, the image is represented by features of a ViT-B/32 [26] Vision Transformer network. Similarly, the text is encoded by a text Transformer [95]. This leads to both image and caption being represented in a 512 dimensional space. To finally assess the alignment of the respective image-text pairs, the cosine similarity is calculated. Based on this, all augmented pairs are re-ranked. In order to provide an intuition on how the CLIP cosine similarity score can be interpreted, selected qualitative examples are visualized in Figure 4.3. Analyzing the augmented data, the previously described failure cases are presented in the left column. Resulting from the weak alignment of the image-text data, the CLIP cosine similarity is low. At the top left, the word *cab* in the sense of *truck cab* is shown on the augmented image patch. However, this token yields the meaning *taxi* which leads to low similarity. Nevertheless, errors caused by tokens with multiple meanings are relatively rare. Instead, errors as shown in the middle left image are much more frequent. Hereby, the bounding boxes of either the original object or the object used for augmentation do not align well. This results in unnatural augmented images showing undesired content. Furthermore, the augmented caption is no longer visually implemented in the image. Another repeatedly occurring error case is represented in the example at the bottom left. Poorly annotated samples of the COCO data include words like *next*, *same* or *another*. Inside the CutSwap dataset pre-processing step, these words are retrieved as an attribute belonging to a corresponding object as the method, up to this point, blindly relies on the annotated data. Unintentionally incorporating such attributes into the augmentation process finally results in poor augmented caption quality. However, the CLIP model is capable of detecting such inaccurate augmentations and accordingly assigns a low image-text similarity. Analyzing augmented data with average and high similarity, no major differences can be observed. Both average similarity as well as high similarity augmented samples constitute meaningful and well aligned image-caption pairs. This leads to the possibility of using a similarity threshold to filter out the poor quality augmentations based on their low similarity scores. Furthermore, the retrieved similarity score is used to re-rank all augmentations above the threshold.

To determine the threshold below which an augmented image-caption pair is discarded, the distribution of CLIP image-text similarity on the CutSwap augmented data is determined using $12\,000$ augmented COCO images. The resulting distribution is shown in Figure 4.4. Hereby, the images result from augmentation and re-ranking where the top-1 image is kept. Qualitative analysis of the data showed that about every tenth augmented pair is of poor quality yielding similarity scores of ~ 0.28 or below. This corresponds well to the determined distribution. Based on these findings, the threshold is set to 0.28.

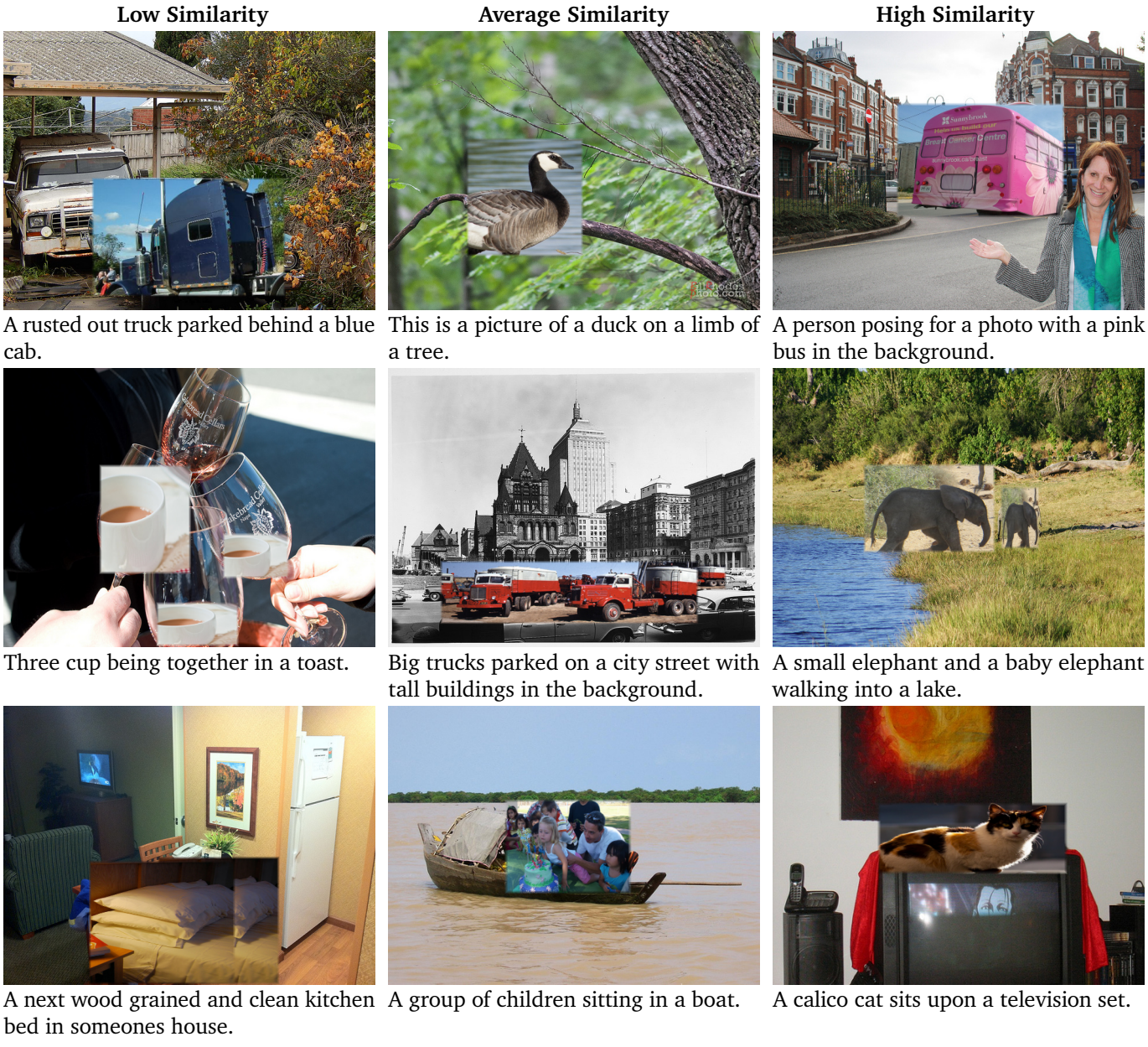


Figure 4.3: CutSwap augmentation examples generated on the COCO dataset. Representative image-caption pairs yielding a low (< 0.26), average (~ 0.30) and high (> 0.34) CLIP cosine similarity are selected based on 1597 examples.

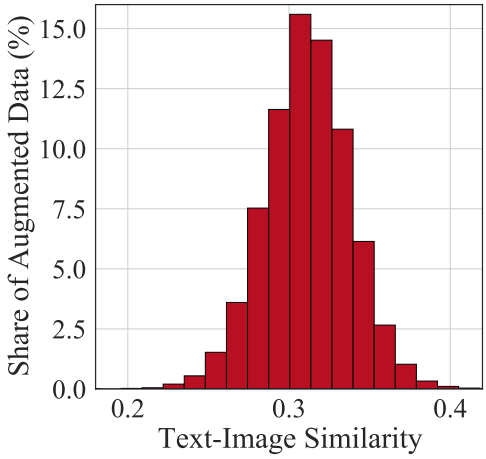


Figure 4.4: CLIP text-image similarity distribution for 12 000 CutSwap augmented images on the COCO dataset.

4.5 Exploring CutSwap Augmented Data

In this chapter, the CutSwap augmented data is analyzed in more detail. The main focus is to examine whether the bottom-up mechanism proposing the input image features is able to function on the augmented images. In addition, the quality of the augmented images is assessed quantitatively. Finally, it is examined whether the characteristics of the augmented data correspond to those of the original COCO data.
It is difficult to directly evaluate the augmented data quantitatively as there does not exist any ground truth data for this task. However, it has been shown qualitatively that CutSwap augmentation results in meaningfull novel data samples. In order to obtain an additional direct quantitative assessment, the FID score, which is commonly used to evaluate newly generated images, is subsequently utilized to evaluate the augmented images. In Table 4.5, the FID scores for LAFITE [113] the state-of-the-art text-to-image generation method on the COCO dataset, the score for the lightweight GAN image manipulation method [52] on COCO and the proposed CutSwap multimodal augmentation are shown. The numbers of the other methods cannot be used for direct comparison but rather indicate in which numerical range other methods operate on the same dataset on similar tasks.
It can be seen that CutSwap performs worse than the generation and manipulation methods in terms of FID. Moreover, the augmented images are composed of real images and should therefore lead to better score by intuition. The reduced score is a result of the globally unnatural properties of the augmented images. Since CutSwap simply inserts a new image patch into the original image, the inevitable result is a globally unnatural image. These unnatural image structures can be observed at the edges of the inserted image section and the context image. Furthermore, they are manifested in different image properties such as colour and perspective of the two image components. Considering the local sections lying outside and inside these edges, the augmented images are obviously of real image quality. Since the FID calculation represents the images through features from one of the last layers of the Inception network, these layers perceive the entire image and accordingly the globally unrealistic property. When captioning the model is fed the associated feature vectors of proposed image regions. More

Table 4.2: Quantitative comparison to image manipulation based of FID on COCO dataset.

Method	**FID** (↓)
LAFITE [113] - *Text-to-Image Generation*	8.12
Lightweight GAN [52] - *Image Manipulation*	8.02
CutSwap	13.79

precisely mean-pooled convolutional feature of the image content inside a bounding box. This results in the globally unrealistic property being of minor importance as long as the bounding boxes enclose appropriate local regions.

4.5.1 Image Feature Extraction

In the following, image feature extraction using the augmented images is investigated. For this purpose, qualitative experiments are conducted. To assess whether the bottom-up attention method can deal with the augmented images, the predicted bounding boxes and the respective predicted visual genome classes are compared between the original and the corresponding CutSwap augmented image. Examples are shown in Figure 4.5. It can be seen that the input feature bounding boxes for both the original and augmented images are quite similar. The bounding boxes in the original context part of the augmented image correspond to those in the original image. Furthermore, the bounding boxes of the swapped image patch align well with the object shown therein. This leads to the conclusion that the region proposal network of the feature extraction method is not disturbed, neither by the image patches nor the resulting edges. In addition, reasonable classes can also be predicted for bounding boxes exceeding the patch, *e. g.* the chair in the bottom right image. More importantly, bounding boxes exceeding the swapped image patch but yielding a wrong class prediction are rare. As for example the *table* bounding box in the top right image. Based on the findings that the bottom-up attention method predicts reasonable bounding boxes and classes, it can be assumed that the resulting input image features are correspondingly reasonable.

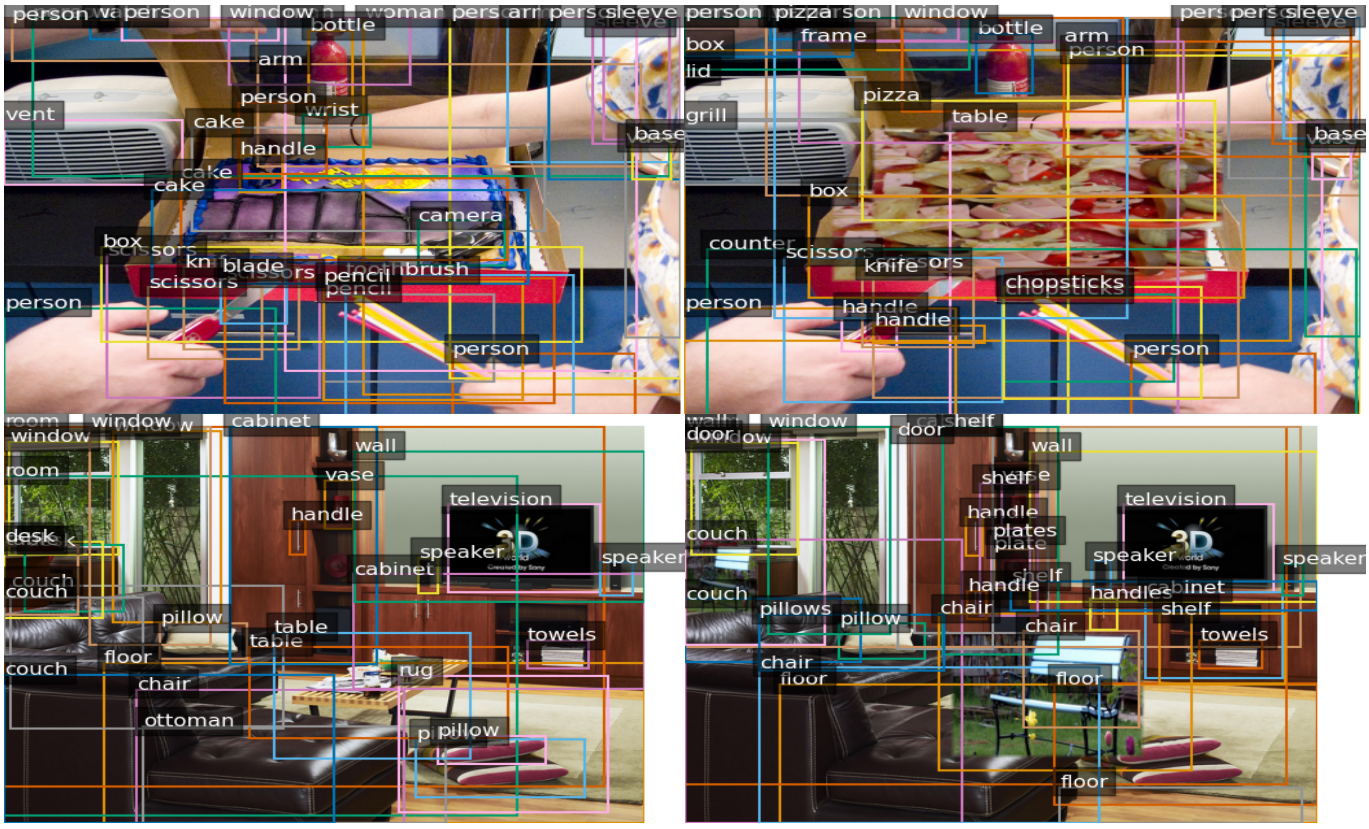


Figure 4.5: Original and CutSwap augmented example images with bottom-up-attention image feature bounding boxes and object class predictions.

4.5.2 Captioning Augmented Images

In order to investigate whether the input image features of the CutSwap augmented data can be reasonably interpreted by the models, captioning on the augmented data is conducted in the following experiment. Hereby, both the UpDown image captioning method and the COS-CVAE diverse image captioning method are examined. In both cases baseline models are used which are only trained on COCO data. Respextive examples are shown in Figure 4.7. It is found that both models can handle the augmented image data as they generate reasonable captions. Compared to the respective CutSwap augmented ground truth captions, the generated sentences can be improved. Nevertheless, both the content of the inserted image patch as well as the context image is properly described by the captions. Based on these results, it can be repeatedly concluded that the globally unnatural properties of the augmented images can be neglected as local parts of the image are used to retrieve the input features. Thus, it is possible to create new combinations of objects and context via the CutSwap augmentation and consecutively use those for training.

CutSwap Augmented Caption
- A zebra walking through the forest in the wild.
- A zebra is walking in the open grass land.
- A zebra grazing a field with trees on a hill.

UpDown
- A zebra standing next to a tree in a field.

COS-CVAE
- A zebra in a grassy field near trees.
- There is a zebra and some trees on the grass.
- A couple of zebra are standing in the grass.

CutSwap Augmented Caption
- Trains parked in front of a library building.
- Red trains passing in front of a building.
- Trains stopped in front of a building.

UpDown
- A red and red train traveling past a building.

COS-CVAE
- A train and a train car parked next to each other.
- A train in front of a large white building.
- A red and red train traveling past a building.

Figure 4.6: Captioning CutSwap augmented images using COS-CVAE and UpDown baseline models. CutSwap augmented captions, the caption predicted by UpDown and random COS-CVAE captioning examples are provided.

4.5.3 CutSwap Data Statistics

The next step is to investigate whether the augmented data follows the characteristics of the COCO data it is based on. The main aim is to ensure that no bias is introduced by the augmentation, for example by amplifying the occurrence of a certain object class. This also verifies that the augmented data reasonably complements the distribution of the original training data so that the captioning models can be optimized on the combined data. Figure 4.7 visualizes the average probability of a certain Visual Genome object class to occur on an image in the dataset for all classes. This is shown for the COCO training data as well as the 12 000 CutSwap augmented images introduced in Sec. 4.4.2. It is observed that the augmented data follows the original coco data regarding the class occurrences. The noisiness of the augmented data distribution is rooted in the fact that 10 times less samples are used for this analysis. Apart from rare outliers, being especially present in the case of classes that rarely appear in the original data set, the distributions are fairly similar.

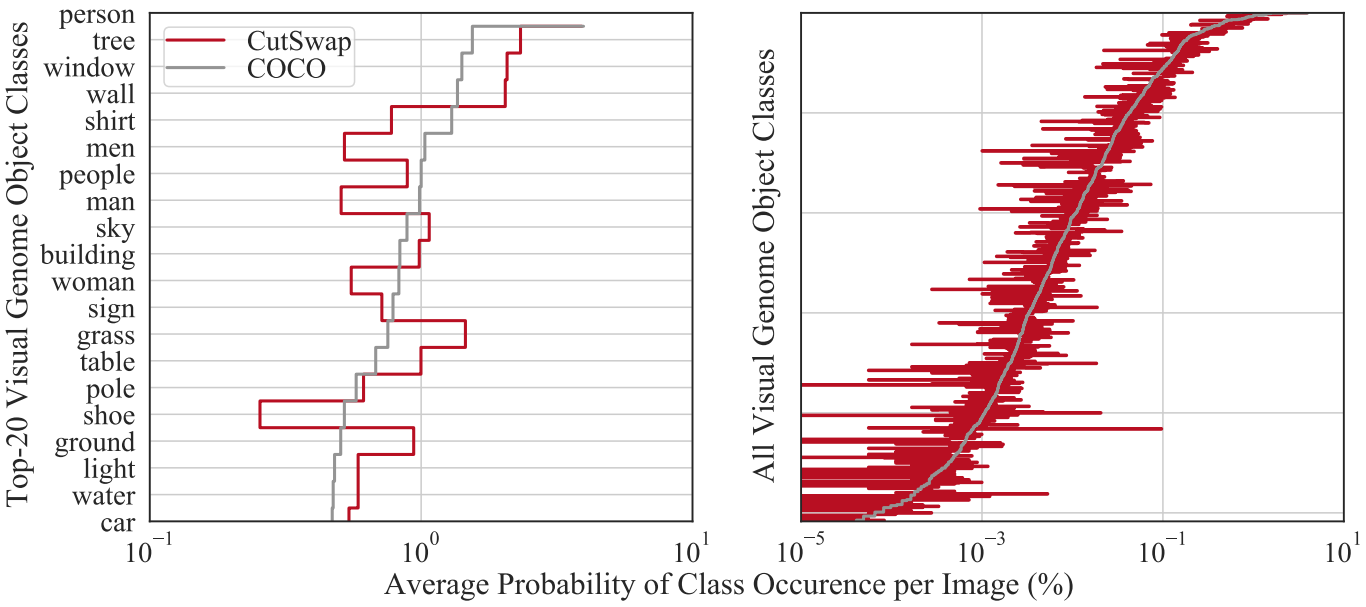


Figure 4.7: Average probability of object class occurrence per image. Visualized for COCO dataset and 12 000 CutSwap augmented images.

4.6 Training with CutSwap Augmentation

Based on the previously discussed experiments, it is shown that CutSwap is able to generate meaningful new data. Furthermore, the augmented data can be interpreted well by the captioning models despite the globally unnatural characteristics of the visual modality. Therefore, the next step is to investigate how enriching the training dataset with CutSwap augmented data affects the performance of the image captioning models. For this purpose, the UpDown image captioning model as well as the COS-CVAE diverse image captioning model are trained on the COCO dataset using CutSwap augmentation. Moreover, experiments are performed on different scaled down sets of COCO. In order to investigate the effect of CutSwap augmentation on the generalization properties, evaluation on the Nocaps dataset is carried out.

All experiments are performed in a strict augmentation scenario. Hereby, the original training setup of the two models is followed rigorously. This results in the amount of training data being the only difference between baseline and CutSwap augmented training. While the baseline models are trained on the COCO dataset, models using CutSwap augmentation extend the COCO data by $10\,\%$, $20\,\%$ or $30\,\%$. All hyperparameters of the two setups are identical and, most importantly, the exact same number of model updates is conducted. This results in the fact that any differences in captioning performance can be directly traced back to the additional CutSwap data. However, this constitutes a very challenging setup since each model update on the augmented data results in having one model update less using the high quality human annotated data.

Following the updated implementation and training schedule [1] of UpDown [4], the model is trained for 70000 iterations. Hereby, a batch-size of 150 image-caption pairs is used and a simple learning rate schedule decreasing linearly from 0.015 to zero is applied. Momentum is set to 0.9 and a weight decay of 0.001 is used. Following [59], the original training setup of the COS-CVAE corresponds to the one described above. Both models are trained on the $118\,287$ COCO images contained in the 2017 train split. Whilst UpDown is evaluated on the entire 5000 images of the 2017 validation split, COS-CVAE is evaluated on a 1000 image subset. Likewise, the CutSwap augmented data is computed using the COCO train 2017 split. Hereby, the augmented data is pre-computed using $n_p = 32$ augmented captions, $n_i = 4$ manipulated images, a similarity threshold of $t_sim = 0.28$ and keeping at most the top-3 re-ranked samples per augmentation. This leads to $44\,289$ augmented image-caption pairs from which all used shares of augmented data are randomly sampled. The aim of this is to reduce the computation time since the same operation would be executed multiple times when conducting the following experiments. In the following methods denoted by the model name are the baseline, trained only on COCO data. Furthermore, the reported numbers stem from the respective paper when provided therein. Otherwise, the baseline models were trained and evaluated to obtain the numbers. Methods denoted by *w CutSwap* refer to the model trained by using additional $20\,\%$ CutSwap augmented data as this share resulted in the best overall performance. The results for methods trained using $10\,\%$ and $30\,\%$ additional CutSwap augmented data can be found in Table A1, Table A2, Table A3 and Table A4.

4.6.1 CutSwap Augmented COCO Dataset

In the following, the results of CutSwap augmentation when training on the full COCO training data are evaluated and discussed. Thereby, $24\,000$ augmented images are added which results in $\sim 20\,\%$ more training examples compared to the baseline.

Image Captioning The results for image captioning using the UpDown model on COCO and Nocaps are reported in Table 4.3 and Table 4.4, respectively. For both evaluated datasets the method using CutSwap augmented training data performs on par with the baseline method. Despite a minor improvement in generalization to near-domain Nocaps data, no consistent gain over the baseline can be observed.

Table 4.3: Single-caption accuracy on multiple metrics for UpDown trained without and with additional 20 % CutSwap augmented data on COCO. Evaluated on COCO.

Method	BLEU-4 (↑)	BLEU-3 (↑)	BLEU-2 (↑)	BLEU-1 (↑)	CIDEr (↑)	ROUGE (↑)	METEOR (↑)	SPICE (↑)
UpDown [1]	**0.372**	-	-	**0.770**	**1.162**	-	**0.278**	**0.210**
UpDown w/ CutSwap	0.370	0.476	0.612	**0.770**	**1.162**	0.569	**0.278**	0.208

Table 4.4: Single-caption accuracy on multiple metrics for UpDown trained without and with additional 20 % CutSwap augmented data on COCO. Evaluated on Nocaps.

Method	*in-domain*		*near-domain*		*out-of-domain*		*Overall*	
	CIDEr (↑)	SPICE (↑)	CIDEr (↑)	SPICE (↑)	CIDEr (↑)	SPICE (↑)	CIDEr (↑)	SPICE (↑)
UpDown [1]	**0.781**	0.116	0.577	0.103	**0.313**	**0.083**	0.553	0.101
UpDown w/ CutSwap	0.774	**0.117**	**0.583**	**0.104**	0.308	0.082	**0.555**	**0.102**

Diverse Image Captioning Table 4.5 and Table 4.6 show the results for accuracy and diversity evaluation for diverse image captioning using the COS-CVAE evaluated on the COCO dataset. Similar to the results on the image captioning task, CutSwap leads to a model performing on par with the diverse image captioning baseline. Single metrics of the accuracy evaluation oracle-20 and oracle-100 show minor improvement over the baseline. However, no consistent gain is achieved. Furthermore, a decrease in accuracy regarding the consensus re-ranking evaluation is observed. This can be explained as follows. Similar to the additional context-based pseudo supervision of the COS-CVAE, the augmented data is forming a distribution that is different to the training data distribution. This is rewarded when generalizing to unseen data such as the test set or a different dataset. In contrast, the consensus re-ranking rewards a model for generating captions close to those of the training distribution. The assumption is that the model improves its generalization properties when CutSwap data is used for training and accordingly the learned distribution differs more from the distribution of the plain paired coco training data. This assumption is strengthened by the qualitative evaluation of the model on Nocaps. Here, the model with CutSwap slightly improves compared to the baseline. Qualitative examples for captioning Nocaps images are shown in Figure A3. For diversity evaluation the same conclusion as for oracle accuracy can be drawn. Here, the method performs similar as the baseline.

Before proceeding with the detailed analysis and interpretation of the results, it should be noted that image captioning and diverse image captioning constitute different tasks. Therefore, the implemented approaches and evaluation protocols differ greatly. Accordingly, there is no meaning in comparing the given numbers. The experiments conducted in Sec. 4.5 qualitatively show that CutSwap augmentation results in meaningful new data. This is hereby quantitatively confirmed since the performance of both the image captioning and diverse captioning model trained with CutSwap augmentation corresponds to that of the respective baseline. In this scenario, 20 % fewer model updates are performed on the original training data as these updates are performed on the augmented data instead. This leads to the conclusion that the augmented data has to be of similar quality as the original data. Based on this finding the following question arises: Why does the method not lead to a clear improvement over the baseline?

Table 4.5: Best-1 accuracy for an oracle evaluation and consensus re-ranking using CIDEr. Accuracy on multiple metrics for COS-CVAE trained without and with additional 20 % of CutSwap augmented data. Evaluation on COCO.

Method	Evaluation	BLEU-4 (↑)	BLEU-3 (↑)	BLEU-2 (↑)	BLEU-1 (↑)	CIDEr (↑)	ROUGE (↑)	METEOR (↑)	SPICE (↑)
COS-CVAE [59]	*Oracle-20*	0.500	**0.640**	**0.771**	**0.903**	1.624	0.706	**0.387**	0.295
COS-CVAE w/ CutSwap		**0.530**	**0.640**	0.770	0.901	**1.629**	**0.708**	0.386	**0.303**
COS-CVAE [59]	*Oracle-100*	0.633	**0.739**	**0.842**	0.942	**1.893**	**0.770**	0.450	0.339
COS-CVAE w/ CutSwap		**0.636**	0.738	**0.842**	**0.947**	1.876	0.767	**0.458**	**0.342**
COS-CVAE [59]	*Consensus*	**0.348**	**0.468**	**0.616**	**0.774**	**1.120**	**0.561**	**0.267**	**0.201**
COS-CVAE w/ CutSwap	*Re-Ranking*	0.311	0.425	0.579	0.756	1.065	0.540	0.258	0.193

Table 4.6: Diversity evaluation on at most the best-5 sentences after consensus re-ranking. COS-CVAE trained without and with additional 20 % of CutSwap augmented data. Evaluation on COCO.

Method	Unique (↑)	Novel (↑)	mBLEU (↓)	Div-1 (↑)	Div-2 (↑)	Self-CIDEr (↑)
COS-CVAE [59]	**96.3**	4404	**0.53**	**0.39**	**0.57**	**0.74**
COS-CVAE w/ CutSwap	95.1	**4459**	0.54	**0.39**	**0.57**	0.73

A first assumption is that the COCO test distribution is already represented sufficiently by the training data. Based on this, the augmented data only introduces minor additional information to the model. Given that there are 80 object classes contained in COCO, these are shown in more than 118 000 images described by more than 590 000 captions. In contrast, approximately 4 % of the amount of data in the case of UpDown and less than 1 % in the case of COS-CVAE is used to quantify how well this distribution is learned.
One explanation for the comparatively small improvement of the generalization properties is that the generated data is limited in its novelty. As no additional data is used for augmentation, the distribution of the augmented data has to be close to the COCO data distribution. This is due to the fact that the method mixes samples from the original distribution based on the grouped object classes. Moreover, this means that improving generalization by using this augmentation setup is limited. Nevertheless, there has to be a scenario in which the training distribution is not represented sufficiently good by the available amount of training data so that the additional information introduced through the augmented data closes this gap.

4.6.2 CutSwap Augmented Reduced COCO Dataset

The following experiments are conducted using smaller amounts of COCO training data. The previously used COCO 2017 training split is from now on denoted as 100 % and the in Sec. 4.6.1 reported results are repeated for the purpose of easier comparison. In addition, the COCO 2014 training split consisting of 82 783 images is denoted as 70 %. Furthermore, the COCO 2014 validation split excluding the 5000 images of the 2017 validation split is used. This results in 35 504 images denoted as 30 %. In addition, a 12 000 images split of the latter is used denoted as 10 %. For every share of training data the corresponding amount of 10 %, 20 % and 30 % CutSwap augmented images is randomly sampled from the pre-computed images. It is ensured that the context image used for augmentation is part of the corresponding reduced training data split. However, the image patches inserted by the CutSwap augmentation are drawn from the full COCO 2017 training data. This results in a portion of new visual data that is incorporated into the reduced dataset via the swapped image patches. Furthermore, the training setup is adjusted. Thereby, only the number of iterations

is modified in order to achieve the best performance on the respective baseline model and avoid overfitting. When training on the 70 % split, the identical setup as for the 100 % training data is used. For the training on the 30 % split, the number of iterations is reduced to 30 000. For training with 10 % data, 7 000 model updates are performed. This setup is found empirically and used for both models. For the following experiments the amount of training data has been reduced until observing a major decrease in performance. Figure 4.8 shows the accuracy as a function of the amount of training data for image captioning using UpDown and diverse image captioning using COS-CVAE evaluated on COCO. The CIDEr and BLEU-4 scores are visualized. Furthermore, the oracle-100 evaluation is shown for the COS-CVAE.

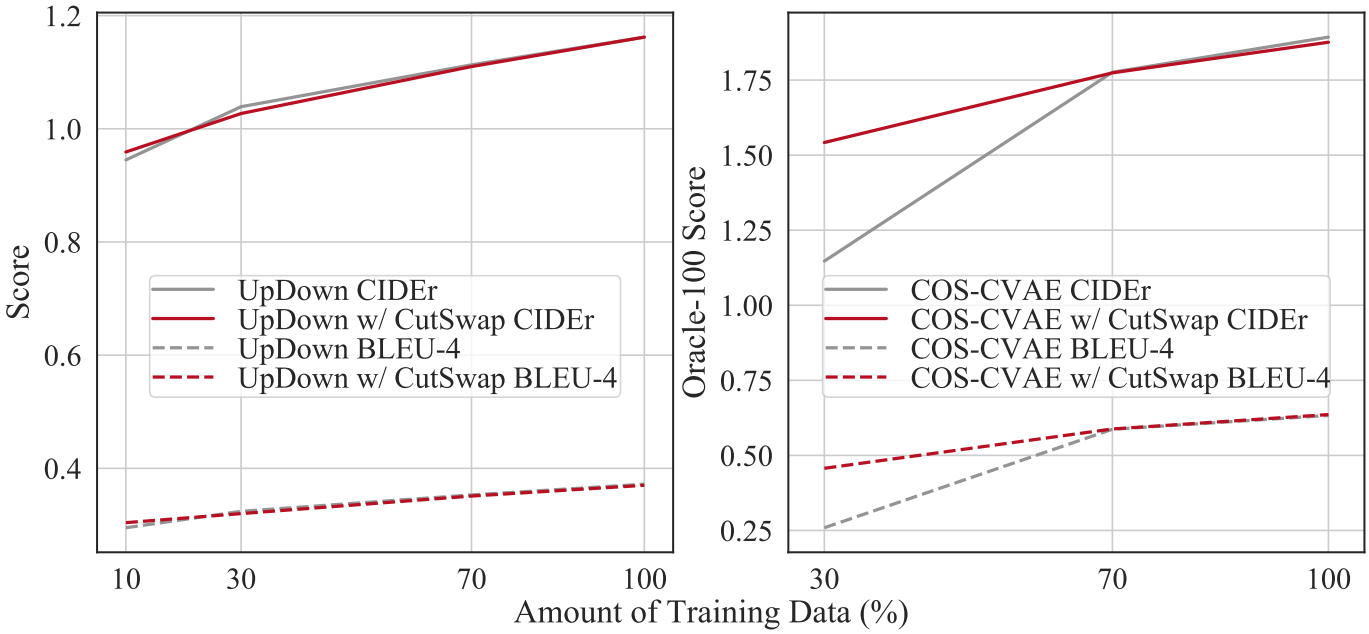


Figure 4.8: UpDown image captioning and COS-CVAE diverse image captioning trained without and with CutSwap on different amounts of COCO training data. Accuracy scores for CIDEr and BLEU-4 on COCO.

Image Captioning Reducing the amount of data when training the UpDown model results in no major performance decrease for 70 % as well as 30 % of data. Only when reducing the training data to 10 %, a noticeable decrease in performance is observed. However, when utilizing only 10 % of the training data, the baseline model maintains 81 % of the performance on COCO and 79 % on Nocaps with respect to the CIDERr metric. For both the 70 % and 30 % split models, the additional CutSwap augmented data does not lead to improvements over the corresponding baseline. The models trained with CutSwap augmentation perform on par or slightly worse. This behavior is observed for the evaluation on the COCO dataset as well as for generalization to Nocaps. The corresponding numbers are reported in Table 4.7 and Table 4.8. Only for the smallest training split of 10 %, augmenting the training data leads to a consistent improvement over the baseline for COCO evaluation. Regarding the generalization to Nocaps, the performance is again in a similar range as the baseline. Based on these results, it is assumed that the capacity of the model is already exhausted when using a fraction of the original training data. Based on this, the predicted captions are examined qualitatively with distinct examples shown in Figure A5. The qualitative performance is analyzed using the 30 % baseline model and the Nocaps dataset. Qualitatively observing the generated captions, it can be noticed that the same generic sentences are used repeatedly. Hereby, single words are exchanged in order to adopt

to the respective image. However, the majority of words as well as the overall sentence structure remains the same for various different images. Furthermore, these generic captions can reasonably describe different images to a certain extend. For images showing people involved in a certain activity the model consistently predicts the caption: *A couple of people that are ...*, and adds the activity tokens at the end of this sentence. In the case of images showing a car in an arbitrary background, the model frequently predicts the caption: *A car is parked in a parking lot.* It can be concluded that the model learns the generic sentence structures using a small fraction of the training data and the additional data only leads to an improved adaptation of the captions to the content shown in the image. Furthermore, this leads to the strong results on the evaluation metrics as repetitive generic captions are not penalized. The UpDown approach conditions the attention and language LSTMs directly on the input image features. Hereby the attention LSTM tends to output a certain word to a certain visual concept which contributes to the observed generic captions. There is no latent space and furthermore no variance is learned as only one caption is generated per image condition. This deterministic setting as well as the limited capacity of Top-Down Attention and language LSTM inevitably leads to generic captions such that various captions describing a similar visual content are not for training. The CutSwap augmentation can be interpreted as enhancing the amount of the data representing a certain underlying concept. Due to the characteristic of the model, this variance of similar data samples cannot be learned. Accordingly, the augmented data only leads to improved results having severely limited amounts of training data, as in the 10 % setting.

Table 4.7: Single-caption accuracy on multiple metrics for UpDown trained without and with additional 20 % of CutSwap augmented data on multiple COCO training data splits. Evaluated on COCO.

Method	**Data Split**	**BLEU-4** (↑)	**BLEU-3** (↑)	**BLEU-2** (↑)	**BLEU-1** (↑)	**CIDEr** (↑)	**ROUGE** (↑)	**METEOR** (↑)	**SPICE** (↑)
UpDown [1]	100 %	**0.372**	-	-	**0.770**	**1.162**	-	**0.278**	**0.210**
UpDown w/ CutSwap		0.370	0.476	0.612	**0.770**	**1.162**	0.569	**0.278**	0.208
UpDown	70 %	**0.353**	**0.461**	**0.599**	**0.759**	**1.113**	**0.560**	**0.271**	**0.204**
UpDown w/ CutSwap		0.351	0.459	0.595	0.757	1.110	0.559	**0.271**	0.203
UpDown	30 %	**0.324**	**0.432**	**0.573**	**0.742**	**1.039**	**0.543**	**0.261**	**0.195**
UpDown w/ CutSwap		0.320	0.429	0.571	0.741	1.027	0.541	0.259	0.194
UpDown	10 %	0.295	0.404	0.547	0.721	0.945	0.524	0.249	0.181
UpDown w/ CutSwap		**0.300**	**0.409**	**0.553**	**0.726**	**0.959**	**0.529**	**0.251**	**0.184**

Table 4.8: Single-caption accuracy on multiple metrics for UpDown trained without and with additional 20 % of CutSwap augmented data on multiple COCO training data splits. Evaluated for generalization to NoCaps validation dataset.

Method	**Data Split**	*in-domain*		*near-domain*		*out-of-domain*		*Overall*	
		CIDEr (↑)	**SPICE** (↑)	**CIDEr** (↑)	**SPICE** (↑)	**CIDEr** (↑)	**SPICE** (↑)	**CIDEr** (↑)	**SPICE** (↑)
UpDown [1]	100 %	**0.781**	0.116	0.577	0.103	**0.313**	**0.083**	0.553	0.101
UpDown w/ CutSwap		0.774	**0.117**	**0.583**	**0.104**	0.308	0.082	**0.555**	**0.102**
UpDown	70 %	**0.761**	**0.114**	0.561	**0.102**	**0.295**	**0.078**	**0.536**	**0.099**
UpDown w/ CutSwap		0.748	0.113	**0.557**	0.101	0.290	**0.078**	0.531	0.098
UpDown	30 %	**0.693**	**0.109**	**0.516**	**0.097**	**0.283**	**0.078**	**0.494**	**0.096**
UpDown w/ CutSwap		0.684	0.108	0.513	0.096	0.277	0.077	0.489	0.095
UpDown	10 %	0.633	**0.104**	**0.457**	**0.091**	**0.243**	**0.074**	**0.439**	**0.090**
UpDown w/ CutSwap		**0.646**	0.103	**0.457**	**0.091**	0.233	0.071	**0.439**	0.086

Diverse Image Captioning The results for diverse image captioning using the COS–CVAE method trained on the different splits for accuracy and diversity evaluation on the COCO dataset are shown in Table 4.9 and Table 4.10, respectively. In addition, the generalization to Nocaps is qualitatively investigated. A random selection of examples is provided in Figure A4. Reducing the amount of training data to $70\,\%$ results in a comparatively small decrease of the evaluated performance. Moreover, the data augmentation still leads to on-par results with the corresponding baseline. Further reducing the amount of training data to $30\,\%$ results in a major drop in performance for both accuracy and diversity. Applying CutSwap augmentation to this setup leads to consistent improvement for all evaluation methods and metrics for both accuracy and diversity. Similarly, the generalization to Nocaps improves noticeably. It can be observed that especially the diversity of the generated sentences is strongly improved by the augmentation. The $30\,\%$ baseline generates very similar sentences in which only single words differ. In contrast, the model trained with augmentation generates distinctly diverse sentences, some of which are entirely different from each other while still accurately describing the image. The results when training on the reduced amounts of data indicate that COS-CVAE learns and benefits from additional data. Reducing the data from $100\,\%$ to $70\,\%$ leads to a comparably small decrease in model performance which is based the assumption derived in Sec. 4.6. The assumption is that the test data is not sufficiently diverse. Furthermore, a comparatively large amount of data describes a restricted amount of visual concepts. This means that above a certain amount of data, no additional information is provided by additional data samples. A further reduction of the data to $30\,\%$ indicates that the captioning method strongly benefits from larger amounts of training data. Compared to the $100\,\%$ setup, for oracle-100 evaluation, only $60\,\%$ of the score on CIDEr is achieved. This can be attributed to the COS-CVAE method and, more precisely, to the fact that a variance is learned in order to generate diverse captions. Hereby, two sequential latent spaces encoding the objects and context of the visual scenes are used for learning a structured latent representation to effectively conditioning the Top-down attention and language LSTM. Thereby, large amounts of diverse data are beneficial since overall a mapping from one-to-many is learned. Correspondingly a multitude of samples describing similar input images is needed to learn die underlying variance of the captions. The COS-CVAE overcomes limited data trough splitting the latent space into context and object and furthermore by leveraging contextual description from the entire dataset for additional supervision. Despite this, having severely limited training data, as in the $30\,\%$ scenario, the amount of data is too limited to obtain a latent space leading to accurate and diverse captions. It is important to note that for every image in the reduced training data, all leveraged captions are still in use for pseudo-supervision. It can be observed that such conditions result in the additional CutSwap augmented data strongly improving the model performance.

4.6.3 Concluding Training Results

In the following, the main findings of the training experiments are summed up, and the most relevant insights are presented. The general idea of the CutSwap multimodal augmentation is to leverage information over the training data in order to extract more information than when directly learning the image-text pairs of the COCO dataset. This is tested for image captioning on the UpDown method and diverse image captioning on COS-CVAE. Using augmentation when training on the entire COCO dataset does not lead to better model performance in either case. On the one hand, the setting chosen for this is highly challenging. On the other hand, the COCO training data appears to already describe the underlying distribution sufficiently. Likewise, the augmentation based on the training data and the underlying CutSwap method results in data samples with a limited amount of novel information. This is the case as the method swaps parts of the Images and corresponding captions within the training data distribution. Regarding the UpDown image captioning approach, the model almost reaches maximum performance by using only a part of the original training data. Due to the deterministic characteristic better performance can hardly result from the additional augmented data. For the task of diverse image captioning using the COS-CVAE the approach already leverages data

Table 4.9: Best-1 accuracy for an oracle evaluation as well as consensus re-ranking evaluation using CIDEr. Accuracy on multiple metrics for COS-CVAE trained without and with CutSwap on multiple COCO splits. Evaluation on COCO.

Method	Data Split	Evaluation	BLEU-4 (↑)	BLEU-3 (↑)	BLEU-2 (↑)	BLEU-1 (↑)	CIDEr (↑)	ROUGE (↑)	METEOR (↑)	SPICE (↑)
COS-CVAE	100 %	*Oracle-20*	0.500	**0.640**	**0.771**	**0.903**	1.624	0.706	**0.387**	0.295
COS-CVAE w/ CutSwap			**0.530**	**0.640**	0.770	0.901	**1.629**	**0.708**	0.386	**0.303**
COS-CVAE	100 %	*Oracle-100*	0.633	**0.739**	**0.842**	0.942	**1.893**	**0.770**	0.450	0.339
COS-CVAE w/ CutSwap			**0.636**	0.738	**0.842**	**0.947**	1.876	0.767	**0.458**	**0.342**
COS-CVAE	100 %	*Consensus Re-Ranking*	**0.348**	**0.468**	**0.616**	**0.774**	**1.120**	**0.561**	**0.267**	**0.201**
COS-CVAE w/ CutSwap			0.311	0.425	0.579	0.756	1.065	0.540	0.258	0.193
COS-CVAE	70 %	*Oracle-20*	0.482	0.626	0.759	0.893	1.574	0.697	0.380	0.290
COS-CVAE w/ CutSwap			**0.482**	**0.627**	**0.763**	**0.896**	**1.587**	**0.700**	**0.383**	**0.294**
COS-CVAE	70 %	*Oracle-100*	0.587	**0.707**	**0.821**	**0.932**	**1.776**	**0.747**	**0.432**	**0.325**
COS-CVAE w/ CutSwap			**0.588**	**0.707**	0.819	0.931	1.774	0.746	0.426	**0.325**
COS-CVAE	70 %	*Consensus Re-Ranking*	0.329	0.443	0.594	**0.763**	1.100	0.547	0.263	0.196
COS-CVAE w/ CutSwap			**0.338**	**0.450**	**0.596**	0.762	**1.115**	**0.552**	**0.265**	**0.197**
COS-CVAE	30 %	*Oracle-20*	0.245	0.405	0.591	0.765	1.118	0.573	0.289	0.214
COS-CVAE w/ CutSwap			**0.389**	**0.558**	**0.709**	**0.858**	**1.416**	**0.656**	**0.342**	**0.262**
COS-CVAE	30 %	*Oracle-100*	0.259	0.421	0.605	0.777	1.147	0.583	0.293	0.220
COS-CVAE w/ CutSwap			**0.457**	**0.617**	**0.753**	**0.892**	**1.542**	**0.688**	**0.375**	**0.285**
COS-CVAE	30 %	*Consensus Re-Ranking*	0.303	0.413	0.562	0.735	0.990	0.531	0.254	0.186
COS-CVAE w/ CutSwap			**0.327**	**0.443**	**0.593**	**0.762**	**1.089**	**0.550**	**0.261**	**0.190**

Table 4.10: Diversity evaluation on at most the best-5 sentences after consensus re-ranking for COS-CVAE trained without and with $20\,\%$ of CutSwap augmented data on multiple COCO splits. Evaluation on COCO.

Method	Data Split	Unique (↑)	Novel (↑)	mBLEU (↓)	Div-1 (↑)	Div-2 (↑)	Self-CIDEr (↑)
COS-CVAE	100 %	**96.3**	4404	**0.53**	**0.39**	**0.57**	**0.74**
COS-CVAE w/ CutSwap		95.1	**4459**	0.54	**0.39**	**0.57**	0.73
COS-CVAE	70 %	**89.5**	4160	0.69	**0.35**	**0.50**	0.64
COS-CVAE w/ CutSwap		89.3	**4195**	**0.68**	**0.35**	0.49	**0.65**
COS-CVAE	30 %	0.31	4001	0.983	0.19	0.21	0.13
COS-CVAE w/ CutSwap		**0.72**	**4147**	**0.825**	**0.31**	**0.40**	**0.51**

across the entire training dataset. This leads to the fact that the additional information provided by the CutSwap data is limited. However, training using CutSwap leads to on par performance with the baseline. Therefore, it can be concluded that the augmented data is of similar quality as the original data. Following this conclusion, there must be a scenario in which the augmentation leads to increased captioning performance. By scaling down the amount of available training data, CutSwap is tested in a low-resource scenario. Hereby, a consistent gain over the baseline on both task and respectively models is achieved. While only a minor increase in performance is achieved for the deterministic Updown model, the stochastic COS-CVAE benefits largely from the additional augmented data. Given a very limited amount of training data, the variance needed to later sample diverse captions cannot be learned since only few data points are available describing similar content. Utilizing additional CutSwap augmented data overcomes this limitation so that diverse captions can be obtained despite having limited training data.

5 Conclusion

5.1 Summary

In the scope of this work, a multimodal data augmentation method for image captioning was developed. Leveraging basic computer vision and natural language processing techniques as well as a multimodal pre-training model led to a lightweight and controllable approach. It is following the key idea to make use of the methods and information already available in image captioning frameworks instead of introducing additional overhead. To the best of our knowledge, CutSwap is the only augmentation method in the context of image captioning performs an augmentation of both modalities while generating meaningful novel data.
In first attempts, text-guided image manipulation GANs were explored for the application in multimodal data augmentation. Since the manipulation performance of modern lightweight methods on complex image data is still limited, the proposed CutSwap augmentation approach was developed. Next, experiments were conducted to determine the design and hyperparameter choices regarding pre-processing of the augmentation data and an image-text similarity based re-ranking approach. This is followed by detailed examination of the augmented data. A key finding was that area-based image encoding solely requires locally natural visual data allowing for images composed of multiple images to be interpreted by captioning models. Finally, the augmentation method was applied for training an image captioning framework and a diverse image captioning framework. Hereby, augmentation was performed within the domain of the training data, which led to mixed results for both approaches. Lastly, the method was tested in a scenario with limited training data. Although minor improvements were achieved in image captioning, the application of the proposed augmentation method for diverse image captioning with limited training data resulted in strongly improved results.

5.2 Future Work

Resulting from the possibility of directly controlling the augmentation process and the properties of the obtained data, a wide variety of possible applications and future research opportunities arise.
Building upon the promising results for diverse image captioning having limited training data, the method can be applied when dealing with low data resources as it is the case for rare languages. Furthermore, the proposed method can be used to obtain augmented data counteracting a certain bias in the training dataset. For example, gender and racial biases in existing datasets can be resolved.
Another promising research direction is using the proposed method to incorporate any visual external data into multimodal dataset. Almost any image dataset can be used by slightly adjusting the data pre-processing step. Thereby, new classes can be integrated into the dataset. Given the design of the method, even unlabeled data can be used since object class and attribute are predicted when obtaining the input image features.

Bibliography

[1] Harsh Agrawal et al. “nocaps: novel object captioning at scale”. In: *Proceedings of the International Conference on Computer Vision*. 2019.

[2] Hiba Ahsan, Nikita Bhalla, Daivat Bhatt, and Kaivankumar Shah. “Multi-Modal Image Captioning for the Visually Impaired”. In: *Annual Meeting of the Association for Computational Linguistics*. 2021.

[3] Peter Anderson, Basura Fernando, Mark Johnson, and Stephen Gould. “SPICE: Semantic Propositional Image Caption Evaluation”. In: *Proceedings of the European Conference on Computer Vision*. 2016.

[4] Peter Anderson et al. “Bottom-Up and Top-Down Attention for Image Captioning and Visual Question Answering”. In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 2018.

[5] Jyoti Aneja, Harsh Agrawal, Dhruv Batra, and Alexander Schwing. “Sequential Latent Spaces for Modeling the Intention During Diverse Image Captioning”. In: *Proceedings of the International Conference on Computer Vision*. 2019.

[6] Martin Arjovsky, Soumith Chintala, and Léon Bottou. “Wasserstein GAN”. In: *International Conference on Machine Learning*. 2017.

[7] Viktar Atliha and Dmitrij Šešok. “Text Augmentation Using BERT for Image Captioning”. In: *Applied Sciences*. 2020.

[8] Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. “Neural Machine Translation by Jointly Learning to Align and Translate”. In: *arXiv 1409.0473*. 2016.

[9] Satanjeev Banerjee and Alon Lavie. “METEOR: An Automatic Metric for MT Evaluation with Improved Correlation with Human Judgments”. In: *Annual Meeting of the Association for Computational Linguistics Workshops on Intrinsic and Extrinsic Evaluation Measures for Machine Translation and/or Summarization*. 2005.

[10] Steven Bird, Ewan Klein, and Edward Loper. “Natural language processing with Python: analyzing text with the natural language toolkit”. In: *O’Reilly Media, Inc.* 2009.

[11] Andrew Brock, Jeff Donahue, and Karen Simonyan. “Large Scale GAN Training for High Fidelity Natural Image Synthesis”. In: *Proceedings of the International Conference on Learning Representations*. 2019.

[12] Tom B. Brown et al. “Language Models are Few-Shot Learners”. In: *Advances in Neural Information Processing Systems*. 2020.

[13] Shashank Bujimalla, Mahesh Subedar, and Omesh Tickoo. “Data augmentation to improve robustness of image captioning solutions”. In: *arXiv 2106.05437*. 2021.

[14] Jiaao Chen, Zichao Yang, and Diyi Yang. “MixText: Linguistically-Informed Interpolation of Hidden Space for Semi-Supervised Text Classification”. In: *Annual Meeting of the Association for Computational Linguistics*. 2020.

[15] Xinlei Chena et al. “Microsoft COCO Captions: Data Collection and Evaluation Server”. In: *arXiv 1504.00325*. 2015.

[16] Alex Clark. “Pillow (PIL Fork) Documentation”. In: *readthedocs*. 2015.

[17] Ekin D. Cubuk, Barret Zoph, Jonathon Shlens, and Quoc V. Le. “RandAugment: Practical automated data augmentation with a reduced search space”. In: *arXiv 1909.13719*. 2019.

[18] Yin Cui et al. “Learning to Evaluate Image Captioning”. In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 2018.

[19] Bo Dai, Sanja Fidler, Raquel Urtasun, and Dahua Lin. “Towards Diverse and Natural Image Descriptions via a Conditional GAN”. In: *Proceedings of the International Conference on Computer Vision*. 2017.

[20] Bo Dai and Dahua Lin. “Contrastive Learning for Image Captioning”. In: *Advances in Neural Information Processing Systems*. 2017.

[21] Michael Denkowski and Alon Lavie. “Meteor Universal: Language Specific Translation Evaluation for Any Target Language”. In: *Annual Meeting of the Association for Computational Linguistics Workshops on Statistical Machine Translation*. 2014.

[22] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. “BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding”. In: *Conference of the North American Chapter of the Association for Computational Linguistics*. 2018.

[23] Jacob Devlin et al. “Exploring Nearest Neighbor Approaches for Image Captioning”. In: *arXiv 1505.04467*. 2015.

[24] Terrance DeVries and Graham W. Taylor. “Improved Regularization of Convolutional Neural Networks with Cutout”. In: *arXiv 1708.04552*. 2017.

[25] Hao Dong, Simiao Yu, Chao Wu, and Yike Guo. “Semantic Image Synthesis via Adversarial Learning”. In: *Proceedings of the International Conference on Computer Vision*. 2017.

[26] Alexey Dosovitskiy et al. “An Image is Worth 16x16 Words: Transformers for Image Recognition at Scale”. In: *Proceedings of the International Conference on Learning Representations*. 2021.

[27] Nikita Dvornik, Julien Mairal, and Cordelia Schmid. “Modeling Visual Context is Key to Augmenting Object Detection Datasets”. In: *Proceedings of the European Conference on Computer Vision*. 2018.

[28] Fartash Faghri, David J. Fleet, Jamie R. Kiros, and Sanja Fidler. “VSE++: Improving Visual-Semantic Embeddings with Hard Negatives”. In: *Proceedings of the British Machine Vision Conference*. 2018.

[29] Steven Y. Feng et al. “A Survey of Data Augmentation Approaches for NLP”. In: *Annual Meeting of the Association for Computational Linguistics*. 2021.

[30] Rinon Gal et al. “StyleGAN-NADA: CLIP-Guided Domain Adaptation of Image Generators”. In: *arXiv 2108.00946*. 2021.

[31] Golnaz Ghiasi et al. “Simple Copy-Paste is a Strong Data Augmentation Method for Instance Segmentation”. In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 2020.

[32] Ian J. Goodfellow et al. “Generative Adversarial Nets”. In: *Advances in Neural Information Processing Systems*. 2014.

[33] Charles R. Harris et al. “Array programming with NumPy”. In: *Nature*. Vol. 585. 7825. 2020, pp. 357–362.

[34] Mareike Hartmann, Aliki Anagnostopoulou, and Daniel Sonntag. “Interactive Machine Learning for Image Captioning”. In: *Proceedings of the National Conference on Artificial Intelligence Workshop on Interactive Machine Learning*. 2022.

[35] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. “Deep Residual Learning for Image Recognition”. In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 2016.

[36] Martin Heusel et al. “GANs Trained by a Two Time-Scale Update Rule Converge to a Local Nash Equilibrium”. In: *Advances in Neural Information Processing Systems*. 2017.

[37] Sepp Hochreiter and Jürgen Schmidhuber. “Long Short-Term Memory”. In: *Neural Computation*. Vol. 9.8. 1997, pp. 1735–1780.

[38] Matthew Honnibal, Ines Montani, Sofie Van Landeghem, and Adriane Boyd. “spaCy: Industrial-strength Natural Language Processing in Python”. In: *to appear*. 2020.

[39] Yifan Jiang, Shiyu Chang, and Zhangyang Wang. “TransGAN: Two Pure Transformers Can Make One Strong GAN, and That Can Scale Up”. In: *Advances in Neural Information Processing Systems*. 2021.

[40] Andrej Karpathy and Fei-Fei Li. “Deep Visual-Semantic Alignments for Generating Image Descriptions”. In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 2015.

[41] Tero Karras et al. “Alias-Free Generative Adversarial Networks”. In: *Advances in Neural Information Processing Systems*. 2021.

[42] Sulabh Katiyar and Samir K. Borgohain. “Image Captioning using Deep Stacked LSTMs, Contextual Word Embeddings and Data Augmentation”. In: *arXiv 2102.11237*. 2021.

[43] Gunhee Kim, Leonid Sigal, and Eric Pà Xing. “Joint summarization of large-scale collections of web images and videos for storyline reconstruction”. In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 2014.

[44] Diederik P. Kingma and Max Welling. “Auto-Encoding Variational Bayes”. In: *Proceedings of the International Conference on Learning Representations*. 2014.

[45] Franz Klein, Shweta Mahajan, and Stefan Roth. “Diverse Image Captioning with Grounded Style”. In: *Proceedings of the German Conference on Pattern Recognition*. 2021.

[46] Sosuke Kobayashi. “Contextual Augmentation: Data Augmentation by Words with Paradigmatic Relations”. In: *Conference of the North American Chapter of the Association for Computational Linguistics*. 2018.

[47] Ranjay Krishna et al. “Visual Genome: Connecting Language and Vision Using Crowdsourced Dense Image Annotations”. In: *arXiv 1602.07332*. 2016.

[48] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E. Hinton. “ImageNet Classification with Deep Convolutional Neural Networks”. In: *Advances in Neural Information Processing Systems*. 2012.

[49] Ashutosh Kumar, Satwik Bhattamishra, Manik Bhandari, and Partha Talukdar. “Submodular Optimization-based Diverse Paraphrasing and its Effectiveness in Data Augmentation”. In: *Annual Meeting of the Association for Computational Linguistics*. 2019.

[50] Yann LeCun, Léon Bottou, Yoshua Bengio, and Patrick Haffner. “Gradient-based learning applied to document recognition”. In: *Proceedings of the IEEE*. Vol. 86. 11. 1998, pp. 2278–2324.

[51] Bowen Li, Xiaojuan Qi, Thomas Lukasiewicz, and Philip H. S. Torr. “ManiGAN: Text-Guided Image Manipulation”. In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 2020.

[52] Bowen Li, Xiaojuan Qi, Philip H. S. Torr, and Thomas Lukasiewicz. “Lightweight Generative Adversarial Networks for Text-Guided Image Manipulation”. In: *Advances in Neural Information Processing Systems*. 2020.

[53] Guodun Li, Yuchen Zhai, Zehao Lin, and Yin Zhang. “Similar Scenes arouse Similar Emotions: Parallel Data Augmentation for Stylized Image Captioning”. In: *Proceedings of the Association for Computing Machinery Multimedia Conference*. 2021.

[54] Xiujun Li et al. “Oscar: Object-Semantics Aligned Pre-training for Vision-Language Tasks”. In: *Proceedings of the European Conference on Computer Vision*. 2020.

[55] Chin-Yew Lin. “ROUGE: A Package for Automatic Evaluation of Summaries”. In: *Annual Meeting of the Association for Computational Linguistics*. 2004.

[56] Tsung-Yi Lin et al. “Microsoft COCO: Common Objects in Context”. In: *Proceedings of the European Conference on Computer Vision*. 2014.

[57] Jiasen Lu, Dhruv Batra, Devi Parikh, and Stefan Lee. “ViLBERT: Pretraining Task-Agnostic Visiolinguistic Representations for Vision-and-Language Tasks”. In: *Advances in Neural Information Processing Systems*. 2019.

[58] Laurens van der Maaten and Geoffrey E. Hinton. “Visualizing Data using t-SNE”. In: *Journal of Machine Learning Research*. Vol. 9. 2008, pp. 2579–2605.

[59] Shweta Mahajan and Stefan Roth. “Diverse Image Captioning with Context-Object Split Latent Spaces”. In: *Advances in Neural Information Processing Systems*. 2020.

[60] Junhua Mao et al. “Deep Captioning with Multimodal Recurrent Neural Networks (m-RNN)”. In: *Proceedings of the International Conference on Learning Representations*. 2015.

[61] Tomas Mikolov et al. “Distributed Representations of Words and Phrases and Their Compositionality”. In: *Advances in Neural Information Processing Systems*. 2013.

[62] Mehdi Mirza and Simon Osindero. “Conditional Generative Adversarial Nets”. In: *arXiv 1411.1784*. 2014.

[63] Seonghyeon Nam, Yunji Kim, and Seon Joo Kim. “Text-Adaptive Generative Adversarial Networks: Manipulating Images with Natural Language”. In: *Advances in Neural Information Processing Systems*. 2018.

[64] Nathan Ng, Kyunghyun Cho, and Marzyeh Ghassemi. “SSMBA: Self-Supervised Manifold Based Data Augmentation for Improving Out-of-Domain Robustness”. In: *Annual Meeting of the Association for Computational Linguistics*. 2020.

[65] Alex Nichol et al. “GLIDE: Towards Photorealistic Image Generation and Editing with Text-Guided Diffusion Models”. In: *arXiv 2112.10741*. 2021.

[66] Maria-Elena Nilsback and Andrew Zisserman. “Automated Flower Classification over a Large Number of Classes”. In: *Indian Conference on Computer Vision, Graphics and Image Processing*. 2008.

[67] Yingwei Pan, Ting Yao, Yehao Li, and Tao Mei. “X-Linear Attention Networks for Image Captioning”. In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 2020.

[68] Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. “BLEU: a method for automatic evaluation of machine translation”. In: *Annual Meeting of the Association for Computational Linguistics*. 2002.

[69] Razvan Pascanu, Tomas Mikolov, and Yoshua Bengio. “On the difficulty of training Recurrent Neural Networks”. In: *International Conference on Machine Learning*. 2013.

[70] Adam Paszke et al. “PyTorch: An Imperative Style, High-Performance Deep Learning Library”. In: *Advances in Neural Information Processing Systems*. 2019.

[71] Or Patashnik et al. “StyleCLIP: Text-Driven Manipulation of StyleGAN Imagery”. In: *Proceedings of the International Conference on Computer Vision*. 2021.

[72] Jeffrey Pennington, Richard Socher, and Christopher Manning. “GloVe: Global Vectors for Word Representation”. In: *Proceedings of the Conference on Empirical Methods in Natural Language Processing*. 2014.

[73] Alec Radford, Luke Metz, and Soumith Chintala. “Unsupervised Representation Learning with Deep Convolutional Generative Adversarial Networks”. In: *Proceedings of the International Conference on Learning Representations*. 2016.

[74] Alec Radford et al. “Learning Transferable Visual Models From Natural Language Supervision”. In: *arXiv 2103.00020*. 2021.

[75] Aditya Ramesh et al. “Zero-Shot Text-to-Image Generation”. In: *arXiv 2102.12092*. 2021.

[76] Scott Reed et al. “Generative Adversarial Text to Image Synthesis”. In: *International Conference on Machine Learning*. 2016.

[77] Shaoqing Ren, Kaiming He, Ross Girshick, and Jian Sun. “Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks”. In: *Advances in Neural Information Processing Systems*. 2015.

[78] David E. Rumelhart, Geoffrey E. Hinton, and Ronald J. Williams. “Learning internal representations by error propagation”. In: *Parallel Distributed Processing - Explorations in the Microstructure of Cognition*. Vol. 1. 1986, pp. 318–362.

[79] Olga Russakovsky et al. “ImageNet Large Scale Visual Recognition Challenge”. In: *International Journal of Computer Vision*. 2015.

[80] Gözde Gül Şahin and Mark Steedman. “Data Augmentation via Dependency Tree Morphing for Low-Resource Languages”. In: *Proceedings of the Conference on Empirical Methods in Natural Language Processing*. 2018.

[81] Tim Salimans et al. “Improved Techniques for Training GANs”. In: *Advances in Neural Information Processing Systems*. 2016.

[82] Mike Schuster and Kuldip Paliwal. “Bidirectional recurrent neural networks”. In: *IEEE Transactions on Signal Processing*. Vol. 45. 11. 1997, pp. 2673–2681.

[83] Rico Sennrich, Barry Haddow, and Alexandra Birch. “Improving Neural Machine Translation Models with Monolingual Data”. In: *Annual Meeting of the Association for Computational Linguistics*. 2016.

[84] Hoo-Chang Shin et al. “Learning to read chest X-rays: Recurrent neural cascade model for automated image annotation”. In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 2016.

[85] Connor Shorten and Taghi M. Khoshgoftaar. “A survey on Image Data Augmentation for Deep Learning”. In: *Journal of Big Data*. 2019.

[86] Ashish Shrivastava et al. “Learning from Simulated and Unsupervised Images through Adversarial Training”. In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 2017.

[87] Karen Simonyan and Andrew Zisserman. “Very Deep Convolutional Networks for Large-Scale Image Recognition”. In: *Proceedings of the International Conference on Learning Representations*. 2015.

[88] Kihyuk Sohn, Honglak Lee, and Xinchen Yan. “Learning Structured Output Representation using Deep Conditional Generative Models”. In: *Advances in Neural Information Processing Systems*. 2015.

[89] Michael Steinbach, George Karypis, and Vipin Kumar. “A Comparison of Document Clustering Techniques”. In: *Proceedings of the International KDD Workshop on Text Mining*. 2000.

[90] Chen Sun, Abhinav Shrivastava, Saurabh Singh, and Abhinav Gupta. “Revisiting Unreasonable Effectiveness of Data in Deep Learning Era”. In: *Proceedings of the International Conference on Learning Representations*. 2017.

[91] Ilya Sutskever, Oriol Vinyals, and Quoc V. Le. “Sequence to Sequence Learning with Neural Networks”. In: *Advances in Neural Information Processing Systems*. 2014.

[92] Christian Szegedy et al. “Going Deeper with Convolutions”. In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 2014.

[93] Christian Szegedy et al. “Rethinking the Inception Architecture for Computer Vision”. In: *arXiv 1512.00567*. 2015.

[94] Ryo Takahashi, Takashi Matsubara, and Kuniaki Uehara. “Data Augmentation Using Random Image Cropping and Patching for Deep CNNs”. In: *IEEE Transactions on Circuits and Systems for Video Technology*. Vol. 30.9. 2020, pp. 2917–2931.

[95] Ashish Vaswani et al. “Attention is all you need”. In: *Advances in Neural Information Processing Systems*. 2017.

[96] Ramakrishna Vedantam, C. Lawrence Zitnick, and Devi Parikh. “CIDEr: Consensus-based Image Description Evaluation”. In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 2015.

[97] Ashwin Vijayakumar et al. “Diverse Beam Search for Improved Description of Complex Scenes”. In: *Proceedings of the National Conference on Artificial Intelligence*. 2018.

[98] Oriol Vinyals, Alexander Toshev, Samy Bengio, and Dumitru Erhan. “Show and Tell: A Neural Image Caption Generator”. In: *International Conference on Machine Learning*. 2015.

[99] Catherine Wah et al. “The Caltech-UCSD Birds-200-2011 Dataset”. In: *Computation & Neural Systems Technical Report: CNS-TR-2011-001*. 2011.

[100] Cheng Wang, Haojin Yang, Christian Bartz, and Christoph Meinel. “Image Captioning with Deep Bidirectional LSTMs”. In: *ACM International Conference on Multimedia*. 2016.

[101] 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: *Advances in Neural Information Processing Systems*. 2017.

[102] Qingzhong Wang and Antoni B. Chan. “Describing like humans: on diversity in image captioning”. In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 2019.

[103] Jason Wei and Kai Zou. “EDA: Easy data augmentation techniques for boosting performance on text classification tasks”. In: *Proceedings of the Conference on Empirical Methods in Natural Language Processing and the International Joint Conference on Natural Language Processing*. 2019.

[104] Tao Xu et al. “AttnGAN: Fine-grained text to image generation with attentional generative adversarial networks”. In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 2018.

[105] Yiben Yang et al. “Generative Data Augmentation for Commonsense Reasoning”. In: *Conference on Empirical Methods in Natural Language Processing*. 2020.

[106] Xin Yi, Ekta Walia, and Paul Babyn. “Generative adversarial network in medical imaging: A review”. In: *Medical Image Analysis*. Vol. 58. 2019, p. 101552.

[107] Zhou Yu, Jing Li, Tongan Luo, and Jun Yu. “A PyTorch Implementation of Bottom-Up-Attention”. In: *https://github.com/MILVLG/bottom-up-attention.pytorch*. 2020.

[108] Sangdoo Yun et al. “CutMix: Regularization Strategy to Train Strong Classifiers With Localizable Features”. In: *Proceedings of the International Conference on Computer Vision*. 2019.

[109] Han Zhang et al. “Stack-GAN: Text to photo-realistic image synthesis with stacked generative adversarial networks”. In: *Proceedings of the International Conference on Computer Vision*. 2017.

[110] Hongyi Zhang, Moustapha Cisse, Yann N. Dauphin, and David Lopez-Paz. “mixup: Beyond Empirical Risk Minimization”. In: *Proceedings of the International Conference on Learning Representations*. 2018.

[111] Pengchuan Zhang et al. “VinVL: Revisiting Visual Representations in Vision-Language Models”. In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 2021.

[112] Xiang Zhang, Junbo Zhao, and Yann LeCun. “Character-level Convolutional Networks for Text Classification”. In: *Advances in Neural Information Processing Systems*. 2015.

[113] Yufan Zhou et al. “LAFITE: Towards Language-Free Training for Text-to-Image Generation”. In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 2022.

[114] Jun-Yan Zhu, Taesung Park, Phillip Isola, and Alexei A. Efros. “Unpaired Image-to-Image Translation using Cycle-Consistent Adversarial Networks”. In: *Proceedings of the International Conference on Computer Vision*. 2017.

Appendices

Figure A1: Random CutSwap augmentation examples on COCO dataset.

Original	CutSwap Augmentation	
		
• A white volkswagon bug with a skateboard on the luggage rack. • A surfboard sits on a car in a parking lot. • A parked car with a surfboard on top of it. • A white vw bug sitting in a parking space with a surfboard on top of it. • A vintage beetle with a surfboard on the top.	• A surfboard sits on a taxi in a parking lot. • A parked taxi with a surfboard on top of it.	• A surfboard sits on a truck in a parking lot. • A parked truck with a surfboard on top of it.
		
• A giraffe running across a desert plain at a high speed. • A giraffe runs across a plain with nothing else in sight. • A lone giraffe running through a vast open space. • A giraffe gallops through an open terrain. • A giraffe running across a dry grass covered field.	• A zebra running across a desert plain at a high speed. • A zebra runs across a plain with nothing else in sight. • A white zebra running through a vast open space. • A zebra gallops through an open terrain. • A zebra running across a dry grass covered field.	• A panda running across a desert plain at a high speed. • A panda runs across a plain with nothing else in sight. • A black panda running through a vast open space. • A panda gallops through an open terrain. • A panda running across a dry grass covered field.

- A couple standing under an umbrella on a beach.
- A man and a woman stand underneath an umbrella.
- A few people that are walking on a beach.
- A couple people standing under an umbrella on the beach.
- People standing on a beach with an umbrella.

- A many children that are walking on a beach.
- A couple children standing under an umbrella on the beach.
- Children standing on a beach with an umbrella

- A elderly people that are walking on a beach.
- A couple people standing under an umbrella on the beach.
- People standing on a beach with an umbrella.

- A man sitting at a table with four glasses of wine in front of him.
- A person sitting at a table with glasses of wine.
- A man is holding a paper while sitting in front of wine.
- Man with four glasses of wine in front of him holding a sheet of paper.
- A man holding a piece of paper with wine in front of him.

- A man sitting at a bed with four glasses of wine in front of him.
- A person sitting at a bed with glasses of wine.

- A man sitting at a bench with four glasses of wine in front of him.
- A person sitting at a bench with glasses of wine.

- A large tiger stripped cat sitting on top of a purse.
- A cat stretches it 's legs on a purse.
- A cat sitting on top of a closed bag.
- A cat laying on a purse on a desk.
- Striped cat sitting on top of a brown purse.

- A large tiger stripped cat sitting on top of a backpack.
- A cat stretches it 's legs on a backpack.
- A cat laying on a backpack on a desk.
- Striped cat sitting on top of a red backpack.

- A large tiger stripped cat sitting on top of a backpack.
- A cat stretches it 's legs on a backpack.
- A cat laying on a backpack on a desk.
- Striped cat sitting on top of a blue backpack.

Figure A2: Example CutSwap augmentation failure cases on COCO dataset.

Original	CutSwap Augmentation	
		
• A break room has a table phone lamp sink and other appliances. • A nice clean looking kitchen with wood fronts and a black dishwasher. • A small break room with a floor lamp. • A small office kitchen has a basket on the table. • A room with a kitchen set up and a calendar on the wall.	• A break room has a bed phone lamp sink and other appliances. • A small office kitchen has a basket on the bed.	• A break room has a bench phone lamp sink and other appliances. • A small office kitchen has a basket on the bench.
		
• A cat looking out of a vehicle window during rain. • A cat sitting on the side of a car door window. • A cat ridding in a car looking out the window with a small building in the background. • A cat looking out a window of a car. • The cat was sitting in the car looking out at the rain.	• A cat was sitting in the truck looking out at the rain. • A cat sitting on the side of a truck door window. • A cat ridding in a truck looking out the window with a small building in the background. • A cat looking out a window of a truck. • The cat was sitting in the truck looking out at the rain.	• A cat was sitting in the cab looking out at the rain. • A cat sitting on the side of a cab door window. • A cat ridding in a cab looking out the window with a small building in the background. • A cat looking out a window of a cab. • The cat was sitting in the cab looking out at the rain.

- A red cup holds dark liquid as it sits next to an open laptop keyboard.
- A red cup of coffee next to an open laptop computer.
- A computer next to a cup of black stuff.
- A cup of black coffee is next to a lap top.
- A cup of black coffee next to a laptop computer.

- A empty wine holds dark liquid as it sits next to an open laptop keyboard.
- A empty wine of coffee next to an open laptop computer.
- A computer next to a wine of black stuff.
- A wine of black coffee is next to a lap top.
- A wine of black coffee next to a laptop computer.

- A black cup holds dark liquid as it sits next to an open laptop keyboard.
- A black cup of coffee next to an open laptop computer.
- A computer next to a cup of black stuff.
- A cup of black coffee is next to a lap top.
- A cup of black coffee next to a laptop computer.

- A purple flower with a stem in a vase on a bowl.
- A large purple flower in a green vase.
- A purple is in a yellow pot.
- A vase sitting in a bowl with a purple flower in it.
- A brightly colored flower blooming from a vase sitting in a bowl.

- A purple flower with a stem in a pitcher on a bowl.
- A large purple flower in a young pitcher.
- A pitcher sitting in a bowl with a purple flower in it.
- A brightly colored flower blooming from a pitcher sitting in a bowl.

- A purple flower with a stem in a vase on a bowl.
- A large purple flower in a grey vase.
- A vase sitting in a bowl with a purple flower in it.
- A brightly colored flower blooming from a vase sitting in a bowl.

- A bench dedicated to someone with a weird edge.
- A wooden park bench on the side of a trail.
- An unoccupied park bench in the woods with a memorial plaque.
- An empty wooden bench in a sunny wood contains two small labels.
- A wooden bench in the dirt by a forest.

- A bench dedicated to someone with a weird edge.
- A wooden park bench on the side of a trail.
- An bright park bench in the woods with a memorial plaque.
- An bright bench in a sunny wood contains two small labels.
- A bright bench in the dirt by a forest.

- A table dedicated to someone with a weird edge.
- A wooden park table on the side of a trail.
- An long park table in the woods with a memorial plaque.
- An long table in a sunny wood contains two small labels.
- A long table in the dirt by a forest.

Figure A3: Qualitative captioning examples on Nocaps dataset for COS-CVAE trained without and with additional 20 % of CutSwap augmented data on the full COCO training data.

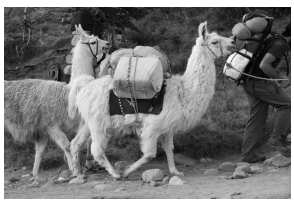
	COS-CVAE	COS-CVAE w/ CutSwap
	• A dog laying down with a person on its head. • A dog laying down with a person in its mouth. • A close up of a person laying on the ground with a person in the foreground. • A dog is holding a person laying on the ground. • A brown and white dog laying down in a field.	• A small person is standing on the ground. • A small orange and white dog is looking at the camera. • A a small orange and white cat is on the ground. • A dog looking up while wearing a collar. • A small orange and white cat is sitting on the ground.
	• A man standing on a surfboard in the water. • A man is holding an umbrella in the water. • A boy standing on top of a surfboard holding a paddle. • A man with a paddle and a paddle board. • A man holding a paddle and a paddle.	• A man standing on a surfboard on a lake. • A man standing on a surfboard with a paddle board on the water. • A person standing on a surfboard in the water. • A young man on a surf board holding a paddle. • A man standing on a surfboard with a paddle board on the water.
	• A couple of zebras standing next to each other. • A zebra laying on top of a pile of grass. • A striped zebra and a cat laying on top of a table. • A zebra laying on the back of a zebra sleeping. • A black and white zebra laying next to a zebra.	• A couple of cats laying on top of a wooden floor. • A cat and a cat laying down in a room. • A cat sitting on top of a wooden floor. • A cat laying on top of a blanket next to another cat. • A cat and a cat sleeping on the floor.
	• A horse on a horse is jumping over a fence. • A brown horse with person riding over a horse. • A jockey riding a horse in the air. • A jockey riding a horse over a building. • A horse with person is on a horse jumping on a horse.	• A man in a helmet riding on a horse. • A man with a horse jumps in the air. • A person riding a brown horse in the air. • A man in a helmet is riding a brown horse. • A man rides a horse in a competition.
	• Several horses standing in a field with a backpack. • A large group of sheep in a field. • A herd of animals standing next to each other. • A sheep on a road with a backpack in the background. • A large group of sheep in a field.	• A group of sheep that are walking down a path. • A black and white photo of a sheep and a person on a field. • A group of sheep that are walking down a path. • Some sheep are standing in a field with a backpack. • A black and white photo of sheep and a person on a dirt ground.

Figure A4: Qualitative captioning examples on Nocaps dataset for COS-CVAE trained without and with additional 20 % of CutSwap augmented data on the 30 % split of the COCO training data.

	COS-CVAE	COS-CVAE w/ CutSwap
	• A girl and a child on a horse. • A girl and a girl on a horse. • A girl and a girl on a horse. • A girl and a child on a horse. • A girl and a girl on a horse.	• A woman riding a horse and a woman in the background. • Two women are riding a brown horse and a fence. • Two women are riding a brown horse in a fenced area. • A woman riding a horse in an arena. • A woman riding on the back of a brown horse.
	• A large brown bear walking on a rock. • A large brown bear standing on a rock. • A large brown bear walking on a dirt road. • A large brown bear walking on a dirt. • A large brown bear standing on a rock.	• A bear walking down a path in a zoo enclosure • A large bear walking down a dirt path. • A large bear walking down a path near some trees. • A bear walking on a rock in a zoo enclosure. • A bear is walking around a log near some rocks.
	• Here is an upscale shop in the district district. • Here is an upscale shop in the middle district. • Here is an upscale shop in the middle district. • Here is an upscale shop in the district district. • Here is an upscale shop in the middle district.	• A man standing in front of a counter at a restaurant. • A man is standing in front of a counter. • Here is an upscale shop in the district. • A man is standing at a bar in a room. • A man is standing in front of a restaurant.
	• A person on a bike is riding a bike. • A person on a bike riding a bike. • A person on a bike is riding a bike. • A person on a bike riding a bike. • A person on a bike riding a bike.	• A man riding a bike bike down a dirt road. • A person riding a bike down a dirt path. • A person riding a bike in the dirt. • A person riding a bike with a helmet in the background. • A person riding on the back of a dirt bike.
	• A bunch of different types of food inside a store. • A bunch of different types of food inside of a store. • A bunch of different types of food inside a store. • A bunch of different types of food inside of a store. • A bunch of different types of food inside a store.	• A store with lots of different types of food. • A display of containers are lined up on a shelf. • A store with lots of different types of food. • There are many different types of food in a store. • A display case with lots of different types of food.

Figure A5: Captioning examples using the UpDown model trained on the 30 % split of the COCO training data. Evaluated on the Nocaps dataset. Selected examples show that UpDown predicts generic sentences for various images.

A car is parked in a parking lot.

A car is parked in a parking lot.

A car is parked in a parking lot.

A car is parked in a parking lot.

A woman in a white shirt playing in the snow.

A woman in a white shirt is playing something.

A woman in a white dress holding a pink flower.

A woman in a white shirt is holding a frisbee.

A group of young men playing a game of soccer.

A group of men are playing a game of frisbee.

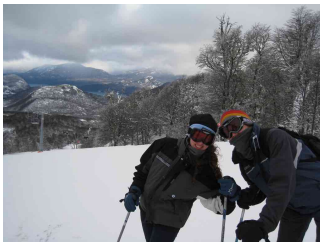
A group of men playing a game of tennis.

A group of men playing a game of soccer.

A couple of people that are in the air.

A couple of people that are sitting on a bench.

A couple of people that are standing on the water.



A couple of people that are standing in the snow.

Table A1: Single-caption accuracy on multiple metrics for UpDown trained without and with CutSwap on multiple COCO training data splits for additional 10 %, 20 % and 30 % of augmented data. Evaluated on COCO.

Method	Data Split	BLEU-4 (↑)	BLEU-3 (↑)	BLEU-2 (↑)	BLEU-1 (↑)	CIDEr (↑)	ROUGE (↑)	METEOR (↑)	SPICE (↑)
UpDown [1]	100 %	0.372	-	-	0.770	1.162	-	0.278	0.210
UpDown w/ 10 % CutSwap		0.369	0.475	0.611	0.769	1.149	0.568	0.276	0.208
UpDown w/ 20 % CutSwap		0.370	0.476	0.612	0.770	1.162	0.569	0.278	0.208
UpDown w/ 30 % CutSwap		0.369	0.475	0.612	0.771	1.158	0.572	0.277	0.209
UpDown	70 %	0.353	0.461	0.599	0.759	1.113	0.560	0.271	0.204
UpDown w/ 10 % CutSwap		0.348	0.456	0.595	0.759	1.110	0.559	0.271	0.202
UpDown w/ 20 % CutSwap		0.351	0.459	0.595	0.757	1.110	0.559	0.271	0.203
UpDown w/ 30 % CutSwap		0.349	0.457	0.596	0.758	1.107	0.56	0.271	0.202
UpDown	30 %	0.324	0.432	0.573	0.742	1.039	0.543	0.261	0.195
UpDown w/ 10 % CutSwap		0.324	0.433	0.575	0.744	1.038	0.543	0.261	0.196
UpDown w/ 20 % CutSwap		0.320	0.429	0.571	0.741	1.027	0.541	0.259	0.194
UpDown w/ 30 % CutSwap		0.322	0.432	0.574	0.743	1.037	0.542	0.260	0.195
UpDown	10 %	0.295	0.404	0.547	0.721	0.945	0.524	0.249	0.181
UpDown w/ 10 % CutSwap		0.304	0.412	0.555	0.727	0.959	0.529	0.251	0.184
UpDown w/ 20 % CutSwap		0.300	0.409	0.553	0.726	0.959	0.529	0.251	0.184
UpDown w/ 30 % CutSwap		0.295	0.404	0.549	0.722	0.933	0.525	0.247	0.182

Table A2: Single-caption accuracy on multiple metrics for UpDown trained without and with CutSwap on multiple COCO training data splits for additional 10 %, 20 % and 30 % of augmented data. Evaluated for generalization to NoCaps validation dataset.

Method	Data Split	*in-domain*		*near-domain*		*out-of-domain*		*Overall*	
		CIDEr (↑)	SPICE (↑)	CIDEr (↑)	SPICE (↑)	CIDEr (↑)	SPICE (↑)	CIDEr (↑)	SPICE (↑)
UpDown [1]	100 %	0.781	0.116	0.577	0.103	0.313	0.083	0.553	0.101
UpDown w/ 10 % CutSwap		0.772	0.115	0.580	0.103	0.300	0.079	0.551	0.108
UpDown w/ 20 % CutSwap		0.774	0.117	0.583	0.104	0.308	0.082	0.555	0.102
UpDown w/ 30 % CutSwap		0.760	0.114	0.572	0.103	0.302	0.080	0.544	0.107
UpDown	70 %	0.761	0.114	0.561	0.102	0.295	0.078	0.536	0.099
UpDown w/ 10 % CutSwap		0.741	0.114	0.559	0.102	0.291	0.079	0.531	0.099
UpDown w/ 20 % CutSwap		0.764	0.113	0.557	0.101	0.290	0.078	0.531	0.098
UpDown w/ 30 % CutSwap		0.746	0.112	0.559	0.100	0.285	0.078	0.531	0.098
UpDown	30 %	0.693	0.109	0.516	0.097	0.283	0.078	0.494	0.096
UpDown w/ 10 % CutSwap		0.697	0.109	0.516	0.096	0.273	0.078	0.492	0.095
UpDown w/ 20 % CutSwap		0.684	0.108	0.513	0.096	0.277	0.077	0.489	0.095
UpDown w/ 30 % CutSwap		0.683	0.106	0.519	0.098	0.263	0.076	0.490	0.095
UpDown	10 %	0.633	0.104	0.457	0.091	0.243	0.074	0.439	0.090
UpDown w/ 10 % CutSwap		0.613	0.102	0.456	0.091	0.231	0.073	0.433	0.090
UpDown w/ 20 % CutSwap		0.646	0.103	0.457	0.091	0.233	0.071	0.439	0.086
UpDown w/ 30 % CutSwap		0.625	0.101	0.465	0.092	0.237	0.075	0.442	0.091

Table A3: Best-1 accuracy for an oracle evaluation as well as consensus re-ranking evaluation using CIDEr. Accuracy on multiple metrics for COS-CVAE trained without and with CutSwap on multiple COCO splits for additional 10 %, 20 % and 30 % of augmented data. Evaluation on COCO.

Method	**Data Split**	**Evaluation**	**BLEU-4** (↑)	**BLEU-3** (↑)	**BLEU-2** (↑)	**BLEU-1** (↑)	**CIDEr** (↑)	**ROUGE** (↑)	**METEOR** (↑)	**SPICE** (↑)
COS-CVAE	100 %	*Oracle-20*	0.500	0.640	0.771	0.903	1.624	0.706	0.387	0.295
COS-CVAE w/ 10 % CutSwap			0.501	0.639	0.768	0.900	1.641	0.707	0.388	0.298
COS-CVAE w/ 20 % CutSwap			0.530	0.640	0.770	0.901	1.629	0.708	0.386	0.303
COS-CVAE w/ 30 % CutSwap			0.490	0.634	0.769	0.896	1.622	0.704	0.389	0.294
COS-CVAE	100 %	*Oracle-100*	0.633	0.739	0.842	0.942	1.893	0.770	0.450	0.339
COS-CVAE w/ 10 % CutSwap			0.626	0.734	0.840	0.947	1.866	0.768	0.450	0.340
COS-CVAE w/ 20 % CutSwap			0.636	0.738	0.842	0.947	1.876	0.767	0.458	0.342
COS-CVAE w/ 30 % CutSwap			0.588	0.709	0.823	0.933	1.797	0.749	0.433	0.325
COS-CVAE	100 %	*Consensus Re-Ranking*	0.348	0.468	0.616	0.774	1.120	0.561	0.267	0.201
COS-CVAE w/ 10 % CutSwap			0.314	0.427	0.576	0.752	1.072	0.539	0.258	0.194
COS-CVAE w/ 20 % CutSwap			0.311	0.425	0.579	0.756	1.065	0.540	0.258	0.193
COS-CVAE w/ 30 % CutSwap			0.329	0.444	0.594	0.765	1.111	0.551	0.265	0.196
COS-CVAE	70 %	*Oracle-20*	0.482	0.626	0.759	0.893	1.574	0.697	0.380	0.290
COS-CVAE w/ 10 % CutSwap			0.492	0.635	0.764	0.894	1.589	0.697	0.378	0.292
COS-CVAE w/ 20 % CutSwap			0.482	0.627	0.763	0.896	1.587	0.700	0.383	0.294
COS-CVAE w/ 30 % CutSwap			0.464	0.612	0.749	0.886	1.548	0.691	0.374	0.285
COS-CVAE	70 %	*Oracle-100*	0.587	0.707	0.821	0.932	1.776	0.747	0.432	0.325
COS-CVAE w/ 10 % CutSwap			0.585	0.703	0.818	0.931	1.774	0.746	0.424	0.323
COS-CVAE w/ 20 % CutSwap			0.588	0.707	0.819	0.931	1.774	0.746	0.426	0.325
COS-CVAE w/ 30 % CutSwap			0.558	0.687	0.806	0.924	1.734	0.736	0.417	0.318
COS-CVAE	70 %	*Consensus Re-Ranking*	0.329	0.443	0.594	0.763	1.100	0.547	0.263	0.196
COS-CVAE w/ 10 % CutSwap			0.329	0.442	0.593	0.764	1.101	0.547	0.263	0.196
COS-CVAE w/ 20 % CutSwap			0.338	0.450	0.596	0.762	1.115	0.552	0.265	0.197
COS-CVAE w/ 30 % CutSwap			0.322	0.435	0.585	0.756	1.097	0.547	0.263	0.194
COS-CVAE	30 %	*Oracle-20*	0.245	0.405	0.591	0.765	1.118	0.573	0.289	0.214
COS-CVAE w/ 10 % CutSwap			0.367	0.532	0.693	0.846	0.376	0.642	0.335	0.253
COS-CVAE w/ 20 % CutSwap			0.389	0.558	0.709	0.858	1.416	0.656	0.342	0.262
COS-CVAE w/ 30 % CutSwap			0.295	0.471	0.645	0.805	0.236	0.608	0.312	0.233
COS-CVAE	30 %	*Oracle-100*	0.259	0.421	0.605	0.777	1.147	0.583	0.293	0.220
COS-CVAE w/ 10 % CutSwap			0.426	0.583	0.730	0.872	0.480	0.670	0.356	0.272
COS-CVAE w/ 20 % CutSwap			0.457	0.617	0.753	0.892	1.542	0.688	0.375	0.285
COS-CVAE w/ 30 % CutSwap			0.333	0.509	0.674	0.831	0.320	0.629	0.327	0.249
COS-CVAE	30 %	*Consensus Re-Ranking*	0.303	0.413	0.562	0.735	0.990	0.531	0.254	0.186
COS-CVAE w/ 10 % CutSwap			0.331	0.443	0.591	0.763	0.084	0.55	0.262	0.195
COS-CVAE w/ 20 % CutSwap			0.327	0.443	0.593	0.762	1.089	0.550	0.261	0.190
COS-CVAE w/ 30 % CutSwap			0.313	0.432	0.583	0.751	0.039	0.540	0.257	0.186

Table A4: Diversity evaluation on at most the best-5 sentences after consensus re-ranking for COS-CVAE trained without and with CutSwap on multiple COCO splits for additional 10 %, 20 % and 30 % of augmented data. Evaluation on COCO.

Method	Data Split	Unique (↑)	Novel (↑)	mBLEU (↓)	Div-1 (↑)	Div-2 (↑)	Self-CIDEr (↑)
COS-CVAE [59]	100 %	96.3	4404	0.53	0.39	0.57	0.74
COS-CVAE w/ 10 % CutSwap		94.5	4383	0.54	0.39	0.56	0.72
COS-CVAE w/ 20 % CutSwap		95.1	4459	0.54	0.39	0.57	0.73
COS-CVAE w/ 30 % CutSwap		82.7	3911	0.73	0.34	0.47	0.61
COS-CVAE	70 %	89.5	4160	0.69	0.35	0.50	0.64
COS-CVAE w/ 10 % CutSwap		88.8	4188	0.69	0.35	0.49	0.64
COS-CVAE w/ 20 % CutSwap		89.3	4195	0.68	0.35	0.49	0.65
COS-CVAE w/ 30 % CutSwap		86.9	4132	0.72	034	0.47	0.62
COS-CVAE	30 %	31.2	4001	0.98	0.19	0.21	0.13
COS-CVAE w/ 10 % CutSwap		61.1	4044	0.87	0.28	0.36	0.43
COS-CVAE w/ 20 % CutSwap		72.4	4147	0.83	0.31	0.40	0.51
COS-CVAE w/ 30 % CutSwap		48.7	4174	0.91	0.27	0.32	0.34