Eidgenössische Technische Hochschule Zürich
Swiss Federal Institute of Technology Zurich

Distributed
Computing

RMCL : A Robust Multimodal Contrastive Learning Framework

Master's Thesis

Stanislas Furrer

sfurrer@student.ethz.ch

Distributed Computing Group
Computer Engineering and Networks Laboratory
ETH Zürich

Supervisors:
Zhao Meng
Prof. Dr. Martin Jaggi
Prof. Dr. Roger Wattenhofer

September 2, 2021

Acknowledgements

I believe none of this work would be possible without my teachers' guidance, encouragement, and mentorship. They have each taught me many important lessons and continue to inspire me.

This master thesis was written between March 2021 and August 2021 at the Distributed Computing Laboratory at ETH Zurich in collaboration with the EPFL (Lausanne), NYC (New York), and LMU (Munich) universities. The project offered a fascinating insight into multi modality algorithms, transformer-based architecture, contrastive learning, and robustness optimization. I am convinced that the valuable experience I was able to gather during my work will be of great help in my future professional career.

First of all, I would like to warmly thanks my supervisors Zhao Meng, Prof. Dr. Martin Jaggi, and Yiran Xing, who excellently guided me and helped me during my Master's Thesis. They provided me with great ideas, advice, motivation and, more importantly, taught me to learn from my failures.

I have also had the privilege of working and discussing with many other researchers, including Kangning Liu, Dr. Yunpu Ma, and Zai Shi. They always motivated and challenged me in a very proactive way to get the best possible results.

Last but not least, I want to express my appreciation to Prof. Dr. Roger Wattenhofer for giving me the chance to work on this exciting project.

Stanislas Furrer

Zurich, September 2021

Abstract

Self-supervised learning has grown in popularity due to its capability to avoid the cost of human-annotated labels. One of the classes of methods that have been behind this recent resurgence of self-supervised learning is named contrastive learning. Contrastive learning has become a dominant component in self-supervised learning methods for natural language processing, computer vision, multi-modality alignment, and other domains. It aims to pull together the embedding of an augmented version of a sample and push apart the different sample's embedding. Contrastive learning benefits from strong augmentation. To be specific, stronger augmentations could expose novel patterns of representations that may improve the generalized ability of the method. Apart from self-supervised, the generalized representation can be seen under the scope of robustness. Indeed, Ilyas et al., 2019 [1] have posit that ; *Adversarial vulnerability is a direct result of our models' sensitivity to well-generalizing features in the data.* They presented the robust optimization as a tool for enforcing the generalization aspect of features learned by deep neural networks. Motivated by the potential synergy between robust optimization and multimodal contrastive learning, we present in this paper RMCL; a robust multimodal contrastive learning. The goal is to reinforce the joint representation of an image-text pair by robust contrastive optimization. In RMCL, we attack a multimodal contrastive task to make the joint representation of an image-text pair invariant to perturbation. In this work, we leverage two well-known contrastive frameworks, BarlowTwins. (Zbontar et al., [2]) and MoCo (Chen et al., 2020 [3]), in robust multimodal settings. We pre-trained a multimodal algorithm ViLT (Kim et al., 2021 [4]) with our RMCL framework and analyze the results. We confirm that our framework drives better robustness against image and text attacks while keeping competitive accuracy on downstream classification tasks with in-domain and out-of-domain datasets. Furthermore, our contrastive approach improves the performance of ViLT on image-text retrieval on both in-domain and out-of-domain datasets. Lastly, our experience manifests that BarlowTwins reaches better overall performance over MoCo under our RMCL settings

Contents

Chapter 1

Introduction

“The road Reaches every place, the short cut only once“
- James Richardson [5]

1.1 Motivation & Related works

The AI field has made massive strides in developing AI systems that learn from vast amounts of carefully labeled data in recent years. Researchers have spent tremendous time and effort curating data and carefully labeling it. However, moving forward, it seems impossible to annotate the vast amounts of data with everything that we care about.

Supervised learning is a bottleneck for allowing more intelligent generalist models to do various jobs and gain new abilities without extensive amounts of labeled data. Supervised learning relies on the definition of categories and optimizes its performance with a highly specific accuracy. In particular, it maximizes accuracy without incorporating much prior context about classified classes, the physical world, or other human-related concepts. This task-specific optimization process comes at the cost of generalization (Wolpert et al., 1997 [6]). Beyond the lack of generalization, the supervised paradigm can lead models to produce shortcut solutions that perform well on a typical test set but fail under different circumstances, revealing a mismatch with our intentions (Geirhos et al., 2020 [7]).

Our world is diverse yet profoundly structured, and humans have an uncanny capacity to make sense of it without someone explicitly teaching it. As babies, we learn how the world runs essentially by perception and association. Generalized knowledge or common sense is taken for given for humans and animals but stay a well-known challenge in Ai research since its origin.

Nowadays, the scientific community has found a promising way to approximate such common sense: self-supervised learning (SSL). Self-Supervised learning offers a promising alternative to supervised learning, where the data itself gives the supervision for a learning algorithm. The fundamental idea of the self-supervised task (also referred to as a pre-training task) is to utilize some section of the data or a transformed version of it to produce labels to solve a supervised task. Specifically, self-supervised tasks act as a proxy strategy to learn representations of the data using pseudo labels. These pseudo labels are created automatically based on the attributes found in the data. Nevertheless, the outcome of this created task is habitually dismissed. Instead, we focus on the learned intermediate representation with the hypothesis that this representation can offer excellent semantic and benefit a diversity of useful downstream tasks.

Specially, we may twist pictures randomly and train a model to predict when pictures are twisted and how much. This prediction task is invented, so the performance is insignificant. However, this pre-training method pushes the model to acquire good semantic notions of objects. Indeed, when transferring this knowledge to a downstream task that intends to recognize the same pictures with a diverse twist, the model has learned to identify high-level object sections, such as heads, noses, and the corresponding locations of these elements, rather than restricted patterns. (Gidaris et al., 2018 [8]).

Self-supervised learning is a representation learning task. Representation learning is a method that learns a parametric mapping from the raw input data domain to a feature vector with the expectation of extracting more abstract and valuable concepts. The representation learned by performing the pre-training task can be used as a starting point for downstream supervised tasks (fine-tuning tasks). Generally speaking, fine-tuning solutions projected from more general representations learned from self-supervised tasks lead to robust predictions and better out-of-sample performance.

In natural language modeling, this pre-training-fine tuning paradigm has been widely used for many years with models such as BERT (Devlin et al., 2018 [9]), RobERTa (Liu et al., 2019 [10]), XLM-r (Conneau et al., 2019 [11]) and many others. The default pre-training task for a language model is to predict the next word given the past sequence. Implicitly, predicting missing parts of the text input makes the model learn to interpret the sense of the words, the syntactic character of the words, and the meaning of entire texts. Natural Language model pre-trained and fine-tuning on a specific supervised downstream task yield greater performance than when only trained in a supervised fashion.

In computer vision, Self-supervised learning is quickly filling the gap with the supervised method on large computer vision benchmarks (Chen et al., 2020 [12] ; Chen et al., 2020 [3] ; Grill et al., 2020 [13] ; Zbontar et al., 2021 [2]). The classes of methods that have been behind this novel resurgence of SSL in computer vision follows a paradigm called contrastive learning. The idea behind contrastive learning is surprisingly simple: the model learns to encode images in a lower-dimensional space so that similar images will be close to each other in the low dimensional space at the same time, far away from other images. For example, we may want the representation of cats to be close to other cats and far away from the representation of dogs. This simple idea is based on nothing else than similarity and leads to potent representation. Contrastive methods do not deflate away any rich semantics by projecting onto a single label selected from only a handful of possible subjective categories. Instead, a discriminative method learns good data representation by comparing an individual instance among different samples.

In contrastive learning, data augmentation is the secret, and indeed a most crucial ingredient in making this method works so well (Tian et al., 2020 [14]). Indeed, it allows the models to have different views of the same signal and make the contrastive task much harder. The contrastive learning approach is instilling invariance to data augmentation. The model learns features that are less affected by the rotation, the brightness, and other types of view of the same signal. It gives a strong proxy for real-life semantics. In a sense, when humans are navigating in the real world, they face different views of the same object.

Beyond computer vision alone, the contrastive learning framework has been recently used to align two modalities. Radford et al., 2021 [15] presented CLIP which efficiently learns visual concepts from natural language supervision. CLIP has shown great success in various classification benchmarks without directly optimizing/fine-tuning for the benchmark's performance. It shows that the image and text relation becomes much more representative thanks to their multimodal-contrastive learning task.

The generalization of a model can also be seen under the scope of robustness. The existence of adversarial examples and the fact that they may correspond to flipping predictive features suggests that deep neural networks make predictions based on vastly different features from what humans use or even recognize. Initially, adversarial examples were view as being either a consequence of the input space being high-dimensional (Gilmer et al., 2018 [16]) or attributed to finite-sample phenomena (Tanay et al., 2016 [17]; Schmidt et al., 2018 [18]).

However, Ilyas et al., 2019 [1] have posit that *Adversarial vulnerability is a direct result of our models' sensitivity to well-generalizing features in the data.* They presented the robust optimization as a tool for enforcing the generalization aspect of features learned by deep neural networks. The authors demonstrate that adversarial robustness leads to more human perception-aligned feature representations. After all, the notion of robustness is human-specified. Adversarial training (and more broadly robust optimization) can be thought of as a tool to incorporate anthropocentric prior over the features. Although adversarially robust models tend to attain lower accuracy than their standardly-trained counterparts, recent work suggests that the feature representations of robust models carry several advantages over those of standard models. Especially, Salman et al., 2020 [19] provides evidence that adversarially robust computer vision models transfer better. This founding leverages the idea that adversarial robustness leads to more generalized features representation.

Only recently, researchers begin associating self-supervised learning to adversarial robustness. Chen et al., 2020 [20] has included adversarial training into self-supervision to produce general-purpose robust pre-trained vision models. Inspired by this research, Kim et al., [21] have leveraged the contrastive pretraining task with adversarial learning and obtain comparable robust accuracy comparing to state-of-the-art supervised adversarial learning tasks. Adversarially robust deep learning occurs to be more data-demanding than conventional learning (Schmidt et al., [18]). It is then reasonable to take advantage of unlabeled data using self-supervised learning. Meng et al., 2021 [22] have improved the robustness of the pre-trained language model BERT by leveraging self-supervised contrastive learning with word-level adversarial perturbation.

The above works and researches constitute our motivation to present in this Master Thesis a Robust Multimodal Contrastive learning framework (RMCL). It consists of leveraging the multimodal contrastive task with adversarial learning to gain robustness against adversarial attacks and synonymously enhance the generalization of multimodal algorithms. Our intuition is that we reinforce the joint representation of an image-text pair $\{I_i, T_i\}$ by conducting robust contrastive optimization. Specifically, given an image-text pair $\{I_i, T_i\}$, we produce deformities on each modality of the pair to maximize the contrastive loss of their joint representation such that their semantic alignment became corrupted. Then, we maximize the correlation between clean image-text pair representation and their adversarial equivalents using contrastive learning to obtain joint representations that overcome deformities produced by adversarial perturbations. It results in learning joint V&L representations that are robust upon adversarial attacks. Since our method does not rely on labels, we used an instance-wise formulation of Projected Gradient Descent (PGD) (Madry et al., 2017 [23]) to attack the pixel-space and the Geometric-inspired attacks (Meng et al., 2020 [24]) to generate word-level adversaries.

To confirm the effectiveness of the proposed RMCL, we leverage two well-known contrastive frameworks MoCo (Chen et al., 2020 [3]) and BarlowTwins (Zbontar et al., 2021 [2]) in a robust multimodal setting and validate our methods on benchmark datasets. We used a pre-trained V&L model ViLT (Kim et al., 2021 [4]) as a baseline and compare its robustness and accuracy performance with and without our RMCL methods. ViLT is a convolution-free V&L model that opens the possibility for on-the-fly pixel-space attacks. Indeed, ViLT uses a lightweight and fast embedding of visuals inspired by ViT (Dosovitskiy et al., 2020 [25]). The results reveal that when transferring on downstream task, the ViLT model pre-trained with RMCL achieve better robustness to natural language adversaries and images adversaries on two classification benchmark NLVR2 (Suhr et al., 2018 [26]) and VQA (Goyal et al., 2017 [27]). Furthermore, our task enhances the model performance on retrieval tasks such as image retrieval and text retrieval (Karpathy et al., 2015 [28]). Our framework has the added value to rely on efficient V&L inputs attacks allowing adversarial training for ViLT on adversarial examples generated on the fly during training instead of generated beforehand

1.2 Contribution

In this thesis, we present a Robust Multimodal Contrastive learning (RMCL) framework. Our idea is to reinforce the latent connection between modalities through their adversarial samples. The synergy from Robust optimization and contrastive learning leads to more robust and analogously more general joint-representation of an image-text pair. Specifically, Our key contributions can be summarized as follows :

- We developed the first Robust multimodal contrastive learning framework using visual and textual adversarial samples as augmented views. We use an instance-wise formulation of Projected Gradient Descent (PGD) to attack the pixel-space and the Geometric-inspired attacks to generate word-level adversaries.
- We extend for the first time the Barlow Twins contrastive framework into multimodal settings.
- We tested our RMCL methods with ViLT as V&L model and either MoCo or Barlow Twins as contrastive frameworks in multimodal settings.
- Also, for the first time, we apply instance-wise attacks in the input space of a V&L model. Indeed, our efficient formulation of the attacks together with ViLT allows an on-the-fly generation of adversaries during training.
- We show that thanks to our RMCL pre-training task, the robustness of ViLT has slightly improved when fine-tuned on downstream tasks. Furthermore, our methods conduct the pre-trained model to have better image and text retrieval performances.
- Finally, we offer some improvement of our concept based on our observation.

1.3 Thesis outline

The remainder of this master thesis is organized as follows: In Chapter 2 we outline the theoretical background and explore the different components for understanding our RMCL. We will review and present the strength of the MoCo and Barlow Twins frameworks and depict the multimodal settings with the ViLT architecture. Furthermore, we will present the augmentation and attacked views used in our contrastive methodology. The experience description will be covered in Chapter 3 and treats the different datasets, methods, models, and optimization used. Following the review of the experiences, we will discuss in Chapter

4 about the results. The choice of attack hyperparameters will be covered, and the comparison of robustness and accuracy of the different methods and models will be illustrated. Finally, Chapter 5 offers conclusions and discussions on the conducted experiences and highlights future work that could further extend and improve this work.

Chapter 2

Background

"In actuality, virtually all learning phenomena resulting from direct experiences can occur on a vicarious basis through observation of other people's behaviour and its consequences for them."
- Albert Bandura in Social Learning Theory, 1991 [29]

Contrastive learning is a family of Self-supervised learning tasks that found great success in computer vision and multi modality alignment. (Chen et al., 2020 [12] ; Chen et al., 2020 [3] ; Grill et al., 2020 [13] ; Zbontar et al., 2021 [2]; Radford et al., 2021 [15]) The rest of this chapter is organized as follows:

In this chapter, we will review the main component of contrastive learning and its notable challenges in section 2.1-2.2. Afterward, we will illustrate in section 2.3 the different multimodal architecture and justify our algorithm choice. We will then present in section 2.4 our Robust multimodal Contrastive Learning methods under two contrastive frameworks; MoCo and Barlow Twins. Finally, we will present the type of clean and attack augmentation of both vision and language used in our contrastive task in section 2.5.

2.1 Self-Supervised Learning

Contemporary self-supervised learning methods can approximately be divided into two families of methods : Predictive/Generative (eg. Oord et al., 2016 [30], Lan et al., 2019[31]) and contrastive (eg. Chen et al., 2020 [12] ; Chen et al., 2020 [3]). The predictive method aims to predict any hidden part of the input from any observed or unhidden section of the input. The predictive strategy is essentially employed in NLP, where it is usual to hide part of the sentence and predict the hidden words from the remaining words. Generally, this is done by solving a classification problem over all finite vocabulary and computing a probabilistic score using a softmax function. However, in computer vision, this task can't be easily extended since images/objects are living in a continuous space rather than a discrete space. Alternatively, contrastive methods are used in computer vision.

Unlike generative models, contrastive learning is a discriminative strategy that strives at grouping similar examples closer and different examples far from each other.

2.2 Contrastive Learning

A popular underlying idea that embodies contrastive methods is their intention to learn invariant representations under diverse distortions (also referred to as data augmentations). It is typically accomplished by maximizing the correlation of representations acquired from various transformed variants of a sample using a alternative of Siamese networks (Bromley et al., 1993 [32]). This idea is illustrated in figure 2.1.

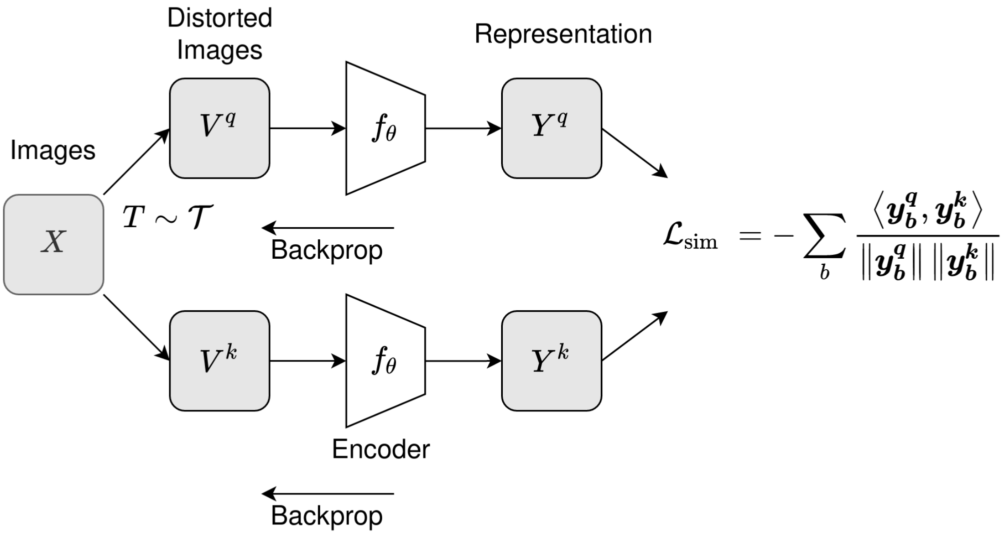

Distorted Images
Representation
Images
V^q
f_θ
Y^q
X
$T \sim \mathcal{T}$
Backprop
$\mathcal{L}_{\text{sim}} = -\sum_b \frac{\langle \boldsymbol{y_b^q}, \boldsymbol{y_b^k} \rangle}{\|\boldsymbol{y_b^q}\| \, \|\boldsymbol{y_b^k}\|}$
V^k
f_θ
Y^k
Encoder
Backprop

Figure 2.1: The underlying structure that unite contrastive methods. Each image of a batch is transformed into two augmented views and fed into an encoder f_θ. Then the model is trained to maximize the similarity of the correlated representations $\boldsymbol{y_q}$ and $\boldsymbol{y_k}$ of the same image. It is usually done by making the cosine similarity between the two vectors as close to one as possible.

We have a batch of sample X, and for each sample, we derive two distorted versions of it by data augmentation. The transformed views are collected by a combination of augmentations $\mathcal{T}$. In particular for images, it could be blur, random cropping and resizing, color distortion or perspective distortion, etc. The two batches of transformed views V_q and V_k are next fed to a function f_θ, ordinarily an encoder with parameters θ, giving batches of representation Y_q and Y_k. Y_q and Y_k have a size of $\mathbb{R}^{n \times m}$ where n is the batch size and m the size of the last hidden layers of f_θ. Next, the model is trained so that two corresponding rows of Y_q and Y_k are close to each other in the embedding space. One way to achieve this purpose is to make the cosine similarity between the two vectors as close to one as possible. The cosine similarity of two variables (vectors) is the cosine of the angle between them and is defined as follows:

$$\cos_{sim}(\boldsymbol{y_q}, \boldsymbol{y_k}) = \frac{\boldsymbol{y_q} \cdot \boldsymbol{y_k}}{\|\boldsymbol{y_q}\| \|\boldsymbol{y_k}\|} \tag{2.1}$$

However, pulling the two vectors together with the cosine similarity alone will lead to a collapse problem where the encoders f_θ will learn to output a trivial constant representation that maximizes the cosine similarity function. In order to prevent this recurrent issue, all the recent approaches have implemented different mechanisms.

2.2.1 SimCLR : The InfoNCE and the projector

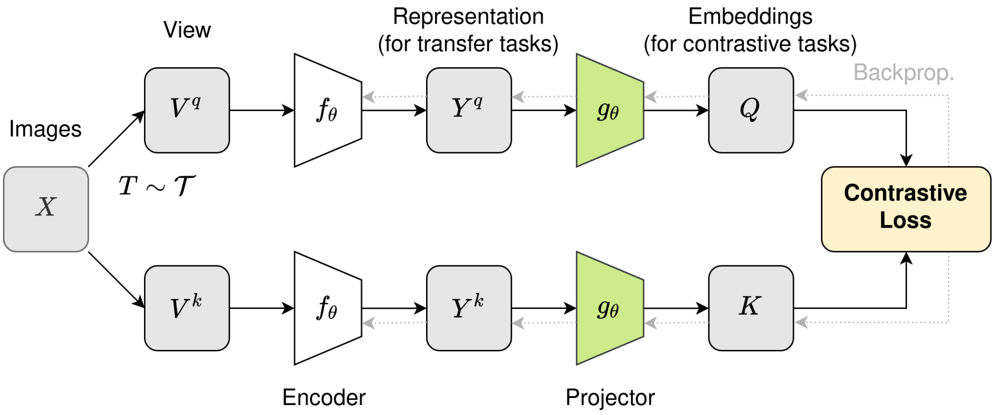


Figure 2.2: SimCLR builds upon the same structure of Figure 2.1 while solving the collapse problem. A projection head g_θ and an encoder f_θ are trained to maximize the similarity using a contrastive loss. Once the training is ended, the projection head g_θ is discarded, and the encoder f_θ together with the representation $\boldsymbol{y_i}$ are used for the downstream tasks.

Contrastive methods like SimClr (Chen et al., 2020 [12]) in figure 2.2 defines “positive” (or similar images) and “negative” (dissimilar images) samples pairs treated differently in the loss function. Throughout the remainder of the thesis, we use the terms of query and key used in (Chen et al., 2020 [3]). We view the similarity matching as a dictionary lookup. We employ the symbols Q (query) as the embedding of Y^q and K (key) as the embedding of Y^k.

SimClr uses the InfoNCE loss (Oord et al., 2018 [33]); a categorical cross-entropy loss to identify the positive sample amongst a set of unrelated noise samples. The loss is defined as followed :

$$L_{infoNCE} = -\log \frac{\exp\left(\mathrm{sim}\left(\boldsymbol{q}, \boldsymbol{k}_{+}\right)/\tau\right)}{\exp\left(\mathrm{sim}\left(\boldsymbol{q}, \boldsymbol{k}_{+}\right)/\tau\right) + \sum_{i=0}^{K_{-}} \exp\left(\mathrm{sim}\left(\boldsymbol{q}, \boldsymbol{k}_{-}^{\boldsymbol{i}}\right)/\tau\right)} \tag{2.2}$$

Where $\boldsymbol{q}$ is the embedding of the original sample, $\boldsymbol{k}_{+}$ represents a positive sample, and $\boldsymbol{k}_{-}$ represents a negative sample. More formally, a key is thought positive $\boldsymbol{k}_{+}$ for a query $\boldsymbol{q}$ if it comes from a similar image and is thought negative $\boldsymbol{k}_{-}$ if it comes from a dissimilar images. The $sim()$ function can be any similarity function, but generally a cosine similarity as defined in equation 2.1 is used. The sensitivity of the product is controls by τ a temperature hyper-parameter. The denominator's sum is calculated over one positive and K_{-} negative pairs in the same batch. It can be seen as a non-parametric variant of $(K+1)$-way softmax classification (Wu et al., 2018 [34]) of $\boldsymbol{q}$ to the corresponding $\boldsymbol{k}_{+}$. When minimizing the InFONCE loss function in equation 2.2, the numerator gauging the similarity of embedding from the matching pair is maximized while the denominator computing the similarity with the dissimilar pairs is minimized.

SimClr's authors have introduced a nonlinear learnable projector g_{θ} between the representations in their framework. Once the contrastive training stage is completed, the projection head $g(f_{\theta})$ is abandoned. Later in transfer learning, the encoder is used as the feature extractor. The authors conjecture that using the representation before the nonlinear projection is due to the suppression of knowledge provoked by the contrastive loss. Indeed, $\boldsymbol{q} = g(f_{\theta})$ is trained to be not affected by data transformation. Therefore g_{θ} may discard beneficial knowledge for the downstream task, such as the color or orientation of objects. This methodology will be kept in the majority of future contrastive frameworks. Additionally, the authors have l2-normalized the embedding vectors $\boldsymbol{q}$ and $\boldsymbol{k}$. Indeed some studies (Wang et al., 2017 [35]; Wang et al., 2020 [36]) have shown the requirement of the norm constraint when doing feature vector dot products in a cross-entropy loss like the InfoNCE loss. (Equation 2.2)

Nevertheless, SimCLR needs a large $4 \sim 8k$ batch size to incorporate enough negative samples to achieve good performance. Contrastive methods tend to work better with larger negative instances since, likely, a bigger amount of negative instances may cover the underlying distribution more efficiently and thus give a better training signal. In SimCLR, the negative keys are from the same batch and updated end-to-end by back-propagation. Since the GPU memory size limits the batch size, the scalability factor with this method remains an issue. Indeed, in its original implementation SimCLR required 32 to 128 TPU v3 cores to train a ResNet-50 with a batch size of 4096.

2.2.2 MoCo_v2 : The Memory Bank and momentum update

Momentum Contrast (MoCo; Chen et al., 2020 [3]) has proposed a solution to this issue by introducing a separate dictionary queue known as a memory bank.

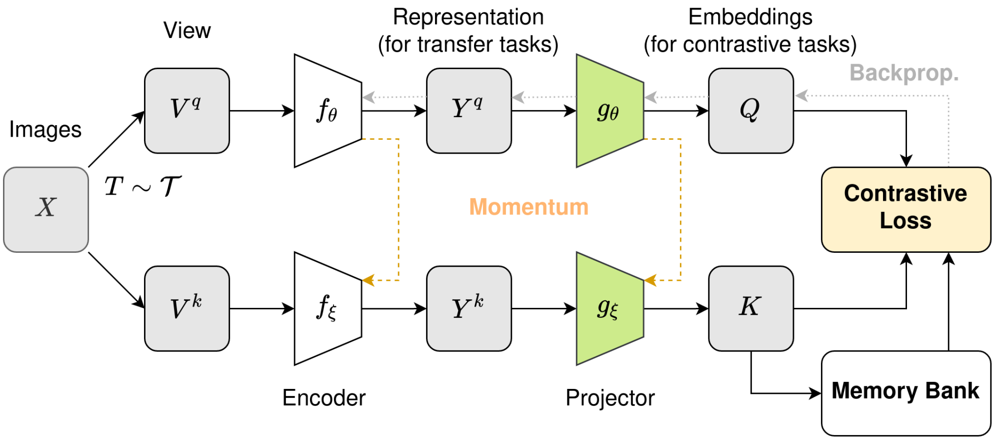


Figure 2.3: MoCo_v2 builds upon the identical structure of Figure 2.1 while solving the collapse problem. The encoder f_θ and a projection head g_θ are trained by pairing an embedded query $\boldsymbol{q}$ to a dictionary of embedded keys using a contrastive loss. The dictionary is a wide FIFO queue accumulating batches of projection $\boldsymbol{k}$ utilized as negative K_- in the contrastive loss. The bottom branch is updated by a momentum update, with the top branch avoiding the intractable computational cost of computing the backpropagation over the very long queue. This method enables the use of a large and consistent memory bank of past projection $\boldsymbol{k}$ as negative examples.

The dictionary is structured as a large FIFO (First-In-First-Out) queue accumulating a large number of projection $\boldsymbol{k}$ (figure 2.3) that are used as negative K_- in the equation 2.2 (The sum in the denominator).As opposed to SimCLR, where the two branches in the illustration designate the identical network (parameterized by θ), MoCo breaks the network into a query network (top line) characterized by θ and a momentum (key) network (bottom line) characterized by ξ. The query network is updated by stochastic gradient descent, while the momentum network is updated based on an exponential moving average of the query network weights. The use of a momentum network enables MoCo to takes advantage of the past projections as negative instance for the contrastive loss. The following equation represents the exponential moving average of the query network weights:

$$\xi \leftarrow m\xi + (1-m)\theta \tag{2.3}$$

In the equation, $m \in [0,1]$ is the momentum coefficient and is set practice very close to 1 (eg. 0.999). Here, ξ indicates the weights of the encoder for negative samples, and θ indicates the weights of the encoder for positive samples. The momentum update avoids the intractable computational cost of computing the backpropagation over the very large queue. Relative to SimCLR, MoCo v2 manages to both decrease the batch size (from 4096 to 256) allowing the framework to run on a typical 8-GPU machine and improve the performance.

2.2.3 BYOL & SimSiam : The asymmetry breaking

In another recent line of work, BYOL (Grill et al., 2020 [13]) and SimSiam (Chen et al., 2021 [37]) have shown that it is possible to solve the collapse problem by only introducing asymmetry in the two branch structure without changing the original cosine similarity loss (equation 2.1).

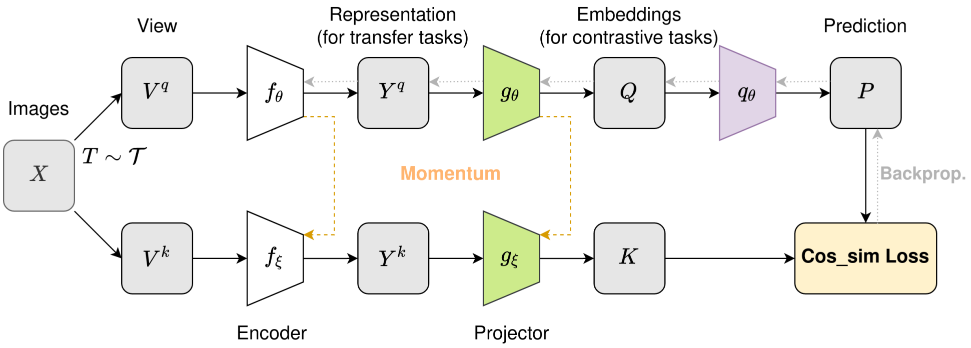


Figure 2.4: BYOL builds upon the same structure of Figure 2.1 while solving the collapse problem. The model parameters are trained by minimizing the row-wise cosine similarity between $q_\theta(Q)$ and K. Indeed, it has been shown that the added projector q_θ is the essential component of this method.(Chen et al., 2021 [37]) Inspired by MoCo_v2 (2.3) the bottom branch is updated by the momentum average of the top branch. The bottom branch being not updated by backpropagation, it is referred to as stop-gradient.

Unlike most popular contrastive learning-based approaches, BYOL and SimSiam do not use negative pairs. As depict in figure 2.4, BYOL use the momentum network concept of MoCo, adding an MLP q_θ (also mentioned as predictor) to predict P from K. BYOL compute the cosine similarity error between the $l2-normalized$ prediction P and target K rather than a contrastive loss. In SimSiam, the authors empirically challenge the necessity of the momentum encoder for preventing collapsing in BYOL. They advocate that only the predictor layer and the stop-gradient operation are critical. In BYOL, the stop-gradient is defined since the momentum encoder isn't explicitly updated by gradient descent but by a momentum update. Unlike BYOL but like SimCLR their method directly shares the encoder's weights between the two branches. The stop-grad consists of not upgrading by backpropagation one of the asymmetric branches.

2.2.4 BarlowTwins : The innovative loss function

Apart from all presented techniques that introduce asymmetries between the two branches (figure 2.1), BarlowTwins (zbontar et al., 2021 [2]) distinguishes itself by its innovative loss function.

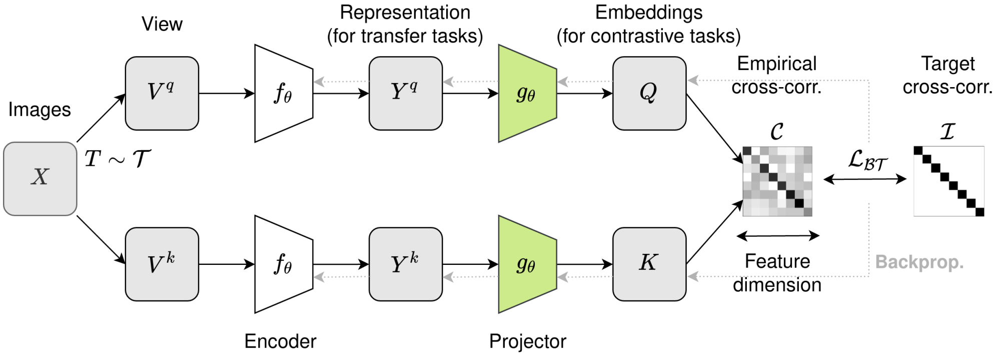


Figure 2.5: BarlowTwins build upon the identical structure of Figure 2.1 while solving the collapse problem. Unlike the other contrastive approach, the BarlowTwins loss function bypasses these trivial solutions by design. Its objective computes the cross-correlation matrix among the embeddings of a pair of equal networks filled with transformed versions of a batch of instances and makes this matrix converge to one. It makes the embedding vectors of transformed versions of instances similar. Furthermore, it minimizes the redundancy among the elements of these vectors.

It does not need any asymmetric mechanisms like momentum encoders, stop-gradients, or prediction networks. The trivial solution is avoided through the design of the loss itself. Like the other methods, BarlowTwins uses a joint embedding of transformed images Q and K, respectively. The loss function is calculated on a correlation matrix, namely C between the two normalized output Q and K of the twin network along the batch dimension :

$$\mathcal{C}_{ij} \triangleq \frac{\sum_b \boldsymbol{q_{b,i}} \boldsymbol{k_{b,j}}}{\sqrt{\sum_b (\boldsymbol{q_{b,i}})^2} \sqrt{\sum_b (\boldsymbol{k_{b,j}})^2}} \tag{2.4}$$

where b indexes the batch sample and i, j index $\boldsymbol{q}$ and $\boldsymbol{k}$ output vectors. C is a squared matrix with dimension the size of the output vectors. Its values are included within -1 (perfect anti-correlation) and 1 (perfect correlation)

Then the loss function $\mathcal{L}_{\mathcal{BT}}$ is defined as followed :

$$\mathcal{L}_{\mathcal{BT}} \triangleq \underbrace{\sum_i (1-\mathcal{C}_{ii})^2}_{\text{invariance term}} + \lambda \underbrace{\sum_i \sum_{j\neq i} \mathcal{C}_{ij}^2}_{\text{redundancy reduction term}} \tag{2.5}$$

where λ is a positive constant trading of the effect of the two terms of the loss. The first term is the *invariante component*. It strives to equalize the diagonal components of the cross-correlation matrix to 1, making the embedding invariant to the transformation applied. The second term is the *redundancy reduction*. It decorrelates the several vector's elements of the embedding by equalizing the off-diagonal components of the cross-correlation matrix to 0. The decorrelation produces the effect of decreasing redundancy between the output so that embedded representations do not include redundant information about the instance. A problem that occurs with non-parametric entropy estimators such as InfoNCE (equation 2.2) is their trend to fall into the curse of dimensionality. Indeed, they are evaluated accurately only in a low-dimensional setting and demand a substantial amount of examples. On the contrary, BarlowTwins can estimate the variability of the embedding from a much smaller batch size and on very large-dimensional embeddings. It is worth noting that BarlowTwins does not normalize the embeddings along the feature dimension as it is the usual method for losses using cosine similarity. Specifically, the features vectors do no longer stay on the unit ball (MoCo; Chen et al., 2020 [3], SimCLR; Chen et al., 2020 [12], BYOL; Grill et al., 2020 [13]).

In the rest of this work, we will use MoCo and BarlowTwins as two different backbones of our Robust multimodal contrastive learning framework. We choose these two frameworks out of the others because first, they are computationally efficient (does not require colossal batch size), and secondly, they embed very different concepts. While both avoid the trivial solution problem, Barlow twins use a unique loss and MoCo, a performant asymmetric method.

2.3 Multimodal framework

Learning general multimodal representations from pictures paired with sentences is crucial for vision-and-language (V&L) tasks. In order to achieve this purpose, several pre-trained V&L models have been introduced recently, motivated by the large-scale pre-training and task-specific fine-tuning methodology found in both computer vision and natural language.

All the V&L pre-training methods aim at producing image-text joint representation from BERT-like objectives. They heavily rely on the self-attention mechanism of Transformers (Vaswani et al., 2017 [38]) to learn pair representations that are properly contextualizing both modalities. The principal distinction among these models comes from the pre-training strategies, the image embedder,

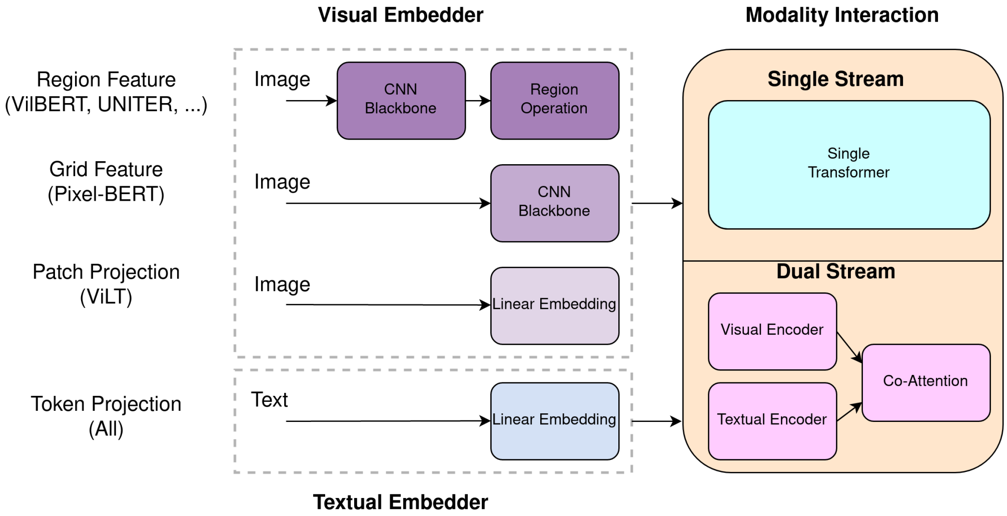


Figure 2.6: A Visual comparison of conventional V&L architectures. Each modality of the image-text pair is feed into its respective encoder. The image and text embedding are then fed to modality interaction modules. The principal distinction between these models comes from the choices of pre-training strategies, the image embedder, and the cross-modality mechanism.

and the cross-modality mechanism. LXMERT (Tan et al., 2019 [39]) and ViLBERT (Lu et al., 2019 [40]) adopted a two-stream approach which consisted of two separate transformer blocks on vision and language embeddings and a third fusion transformer block for cross-modality. UNITER (Chen et al., 2019 [41]) and VisualBert (Li et al., 2019 [42]) employed a single stream of transformer to learn image-text embedding jointly.

To be specific, the figure 2.6 illustrates the structure of the large majority of the V&L models. Given a text-image pair, both modalities are encoded through their respective encoders, and afterward, their cross-modal representation is computed through either a single-stream or a two-stream transformer.

Emanuele et al., 2021 [43], have shown that single and dual stream model mechanism are on similar when evaluated on downstream tasks. However, the embedding layers are essential in a model's final performance.

For a long period, most of the V&L models were using a heavy pre-trained object detector (Faster-RCNN; Ren et al., 2015 [44]) as visual embedder. It extracts the region of interest (RoIs) for a given image and computes the spatial location of its bounding box. Another common approach (X-LXMERT; Cho et al., 2020 [45], Pixel-BERT; Huang et al., 2020 [46]) was to use the output feature grid of convolutional neural networks such as ResNets (He et al., 2016 [47]). However, both these visual embedder methods contribute to the largest portion of the model computation. The weaknesses of holding a massive visual embedder are frequently ignored in academic practices. Indeed, the region features are saved beforehand the training stage to reduce the weight of feature extraction. However, these limitations restrict the real-world deployment as the query on the fly has to face a slow extraction process.

2.3.1 ViLT : The linear projection of a patch

In order to solve this bottleneck, ViLT (Kim et al., 2021 [4]) uses a linear projection that operates on image patches as a visual embedder. Indeed, recent work (Dosovitskiy et al., 2020 [25]) proved that adopting a naive linear projection of a patch is efficient to embed pixels before inputting them inside transformers. ViLT is a single stream convolution-free V&L model that achieves competitive performance while significantly reducing the inference. Unlike the other V&L model that uses BERT as a transformer, ViLT uses ViT (Dosovitskiy et al., 2020 [25]) as a cross-modal transformer.

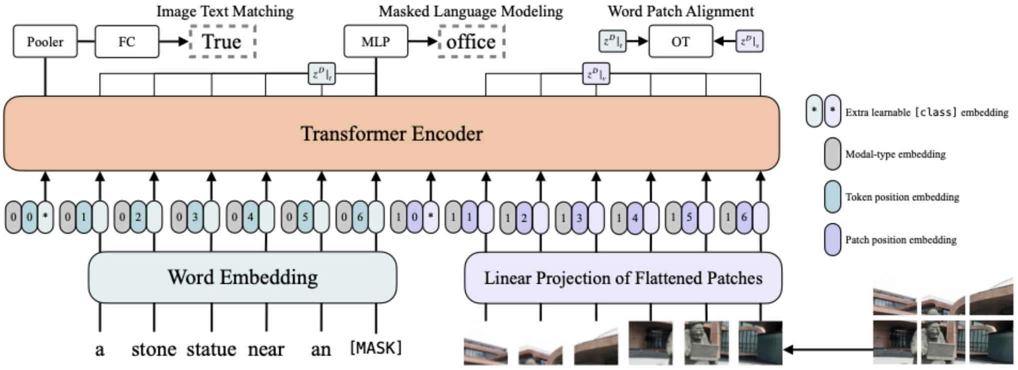


Figure 2.7: ViLT overview. A text is tokenized into tokens, while an image is split into patches and linearly projected. The embedded representation of both modalities is concatenated together with their position. Furthermore, a special token for each modality t_{class} and v_{class} is added to the sequence. The sequence is then fed into a ViT transformer. In addition to the representation of the all sequence Z^D a joint-representation $\boldsymbol{CLS}$ of the all sequence is computed by pooling the output of the encoder. See equation 2.6 for more details

This Master Thesis will use ViLT as a multimodal architecture due to its little inference time and competitive performance. Indeed, we need the shortest inference time since we will attack the pixel and word-level spaces.

ViLT structure is illustrate in fgure 2.7.

$$
\begin{aligned}
\bar{t} &= [t_{\text{class}}\,; t_1T; \cdots ; t_LT] + T^{\text{pos}} \\
\bar{v} &= [v_{\text{class}}\,; v_1V; \cdots ; v_NV] + V^{\text{pos}} \\
z^0 &= \left[\bar{t} + t^{\text{type}}\,; \bar{v} + v^{\text{type}}\,\right] \\
\hat{z}^d &= \text{MSA}\left(\text{LN}\left(z^{d-1}\right)\right) + z^{d-1}, \qquad d = 1 \ldots D \\
z^d &= \text{MLP}\left(\text{LN}\left(\hat{z}^d\right)\right) + \hat{z}^d, \qquad d = 1 \ldots D \\
CLS &= \tanh\left(z_0^D W_{\text{pool}}\,\right)
\end{aligned}
\tag{2.6}
$$

ViLT is a simple V&L architecture that uses a lightweight visual embedder and a single stream approach. It consists of a succession multiheaded self-attention (MSA) layer and an MLP layer.[1]

The input text t consist of L words from the English Vocabulary |V| embedded to $\bar{t} \in \mathbb{R}^{L\times H}$ with a word embedding matrix $T \in \mathbb{R}^{|V|\times H}$ and a location embedding matrix $T^{\text{pos}} \in \mathbb{R}^{(L+1)\times H}$ [2]. An additional special token t_{class} is added to the embedded list $\bar{t}$. The input image $I \in \mathbb{R}^{C\times H\times W}$ is divided into a sequence of Patches. The 2D patches are flattened $v \in \mathbb{R}^{N\times\left(P^2\cdot C\right)}$ where (H, W) is the resolution of the initial image, C is the number of channels, (P, P) is the resolution of each image patch, and $N = HW/P^2$ is the resulting number of patches. Afterwards, v is embedded into $\bar{v} \in \mathbb{R}^{N\times H}$ by linear projection $V \in \mathbb{R}^{\left(P^2\cdot C\right)\times H}$ and location embedding $V^{\text{pos}} \in \mathbb{R}^{(N+1)\times H}$. An additional special token v_{class} is added to the embedded list $\bar{v}$.

The modal embedding vectors t^{type},$v^{type} \in \mathbb{R}^H$ are added to their analogous vision and language embeddings, then are concatenated into a combined sequence z_0. The vector $\boldsymbol{z}$ is computed by D-depth transformer layers up until the last sequence z_d. $\boldsymbol{CLS}$ is a pooled representation of the entire multimodal input and is computed by applying linear projection $W_{\text{pool}} \in \mathbb{R}^{H\times H}$ and hyperbolic tangent on the first index of the sequence z^D. In this work, we denote $\boldsymbol{CLS}$ as the joint representation of the entire image-text pair. The joint representation will be the element we will contrast through our contrastive methodology.

2.3.2 ViLT : The Pre-Training tasks

Originally, ViLT was pre-trained with two regularly used V&L objectives: masked language modeling (MLM) and Image text matching (ITM). (MLM).

Image-Text Matching (ITM) In ITM, the aligned image is randomly substituted with a distinct image with the probability of 0.5. The $\boldsymbol{CLS}$ is projected by an ITM head made of a unique linear layer. It output a binary class that

[1] The unique distinction between ViT and BERT is the location of layer normalization (LN). The layer normaliyation appears before MSA and MLP in ViT and after in BERT.

[2] For both modality, the positions are standard learnable 1D position embeddings

tells whether or not the image matches the text. We denote the output score as $g_\theta(\boldsymbol{CLS})$. Then a binary cross-entropy loss is computed as ITM loss:

$$\mathcal{L}_{\text{ITM}}(\theta) = -\mathbb{E}_{(\mathbf{t},\mathbf{I})\sim D}\left[y \log_\theta(\boldsymbol{CLS}) + (1-y)\log\left(1 - g_\theta(\boldsymbol{CLS})\right)\right]\Big) \tag{2.7}$$

where $\boldsymbol{t}$ is the text and $\boldsymbol{I}$ is the corresponding image. Plus, motivated by the word region alignment objective in Chen et al. 2019 [41], ViLT's authors have design a word patch alignment (WPA) that estimates the matching score between two subsets of z^D : $z^D\big|_t$ (textual subset) and $z^D\big|_v$ (visual subset). They use the inexact proximal point method for optimal transports (IPOT) (Xie et al., 2020 [48]). For example, given a word token it can computes the alignment score with all the patch of the images.

Masked Language Modeling (MLM) The objective aims is to predict a randomly masked word t_{masked} with mask indices $\mathbf{m} \in \mathbb{N}^M$ [3] based on their neighboring words $\mathbf{t}\backslash\mathbf{m}$ [4] and all image patch V. A word t is randomly mask with the probability of 0.15. The MLM loss is mesured as the negative log-likelihood loss for the masked tokens as followed :

$$\mathcal{L}_{\text{MLM}}(\theta) = -\mathbb{E}_{(\mathbf{t},\mathbf{I})\sim D} \log P_\theta\left(\mathbf{t}_\mathbf{m} \mid \mathbf{t}_{\backslash\mathbf{m}}, \mathbf{I}\right) \tag{2.8}$$

where θ is the trainable parameters and D is the training set. We will use the ITM-MLM pre-trained ViLT model as the backbone of our contrastive framework.

2.4 Contrastive learning in Multi-modal settings

In this project, we propose to extend the contrastive approach in the multi-modal settings while using adversaries as augmented views. We believe that pre-training a V&L model by adversarial training on a multimodal contrastive framework will yield a more generalized and robust joint representation of image-text pairs. Specifically, a V&L model pre-trained with our RMCL and fine-tuned on a downstream task should be more robust to adversaries and have better zero-shots performance.

In this current project, we test the following assumption by extending MoCo and BarlowtTwins framework in the multimodal setting while using adversaries as augmented views. It is worth noting that our RMCL framework could be applied to any contrastive methods presented so far. In both frameworks, the architecture is divided into two branches.[5] In MoCo, the two branches are asymmetric, while

[3] M is the number of masked tokens, and m is the set of masked indices.

[4] Notation for : $t_{\backslash m} = \{t_1, \ldots, t_{i-1}, [\text{ MASK }], t_{i+1}, \ldots, t_L\}$

[5] Strictly speaking, BarlowTwins under our settings could be represented as a single branch. However, for the sake of consistency in the notation, we will keep the two branch representation.

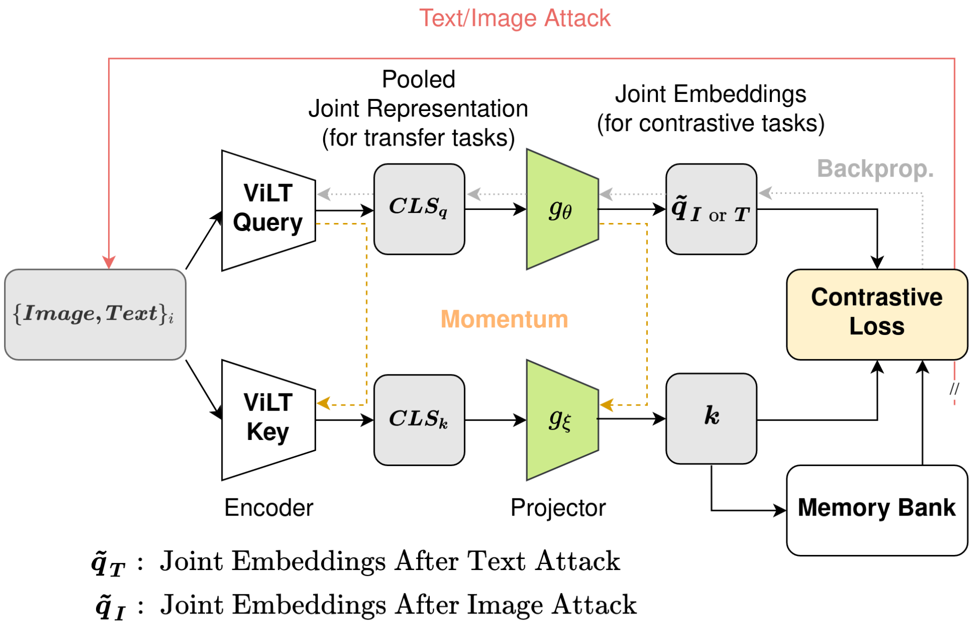


Figure 2.8: RMCL overview under MoCo settings. An Image-text pair is fed into a ViLT encoder, and a pooled vector $\boldsymbol{CLS}$ represents the joint representation of all multimodal input. The joint representation is projected used in the contrastive loss. The RMCL objective is then attacked on the fly independently in the pixel-space and in the words-space. The attacked joint embeddings $\tilde{\boldsymbol{q}}_{\boldsymbol{I}}$ and $\tilde{\boldsymbol{q}}_{\boldsymbol{T}}$ are then contrast with the dissimilar clean joint embeddings $\boldsymbol{k}_-$

in BarlowTwins, they share the same weights. Indeed, in section 2.2 we have seen that in MoCo the bottom branch is updated by the moving average of the top branch weights following the equation 2.3 while a regular SGD-based optimizer updates the top one. In contrast, the two branches of BarlowTwins are updated by an SGD-based optimizer. We use the MoCo notation for both methods and call the top branch the query network and the bottom branch the key network. In both methods, a batch of Images-text pairs from a dataset is fed into the query and key ViLT encoders.

As illustrate in the figure 2.8 $\boldsymbol{CLS}$ refers to the pooled representation of the whole multimodal input (see equation 2.6). The joint representation $\boldsymbol{CLS_q}$ and $\boldsymbol{CLS_k}$ are then fed into their respective Moco or BarlowTwins projector g_θ. Note that the illustration 2.8 shows our framework build on top of MoCo. We designate the output of the encoder the 'joint representations' and the projector's output the 'joint embeddings'. In general, the query embedding is $\boldsymbol{q}$ and the key embedding is $\boldsymbol{k}$. The representations are employed for downstream tasks, and the embeddings are fed to the loss function of the Multimodal MoCo and the multimodal BarlowTwins. In moco, the batch of $\boldsymbol{k}$'s embeddings are saved systematically on the memory bank and used as negative pairs K_-.

Given an image-text pair $\{I_i, T_i\}$, we denote $\tilde{\boldsymbol{q_I}}$ as the joint Embeddings of the pair after image attack and $\tilde{\boldsymbol{q_T}}$ as the joint Embeddings of the pair after text attack. We refer $\boldsymbol{k}_+$ as the joint embedding of the same pair not attacked and $\boldsymbol{k}_-$ as the joint embedding of dissimilar pairs not attacked. Our Robust Multimodal InfoNCE loss with MoCo settings is then defined as follow :

$$L_{RMinfoNCE} = \sum_{\tilde{\boldsymbol{q}} \in \{\tilde{\boldsymbol{q_I}}, \tilde{\boldsymbol{q_T}}\}} - \log \frac{\exp\left(\tilde{\boldsymbol{q}} * \boldsymbol{k}^+ / \tau\right)}{\exp\left(\tilde{\boldsymbol{q}} * \boldsymbol{k}^+ / \tau\right) + \sum_{K^-} \exp\left(\tilde{\boldsymbol{q}} * \boldsymbol{k}^- / \tau\right)} \tag{2.9}$$

where τ is a temperature hyper-parameter. The first term of the sum aim to pull together the joint-embedding of a given pair and its counterpart attacked in the pixel space while pushing apart dissimilar clean pairs. The second term does the same but for the joint-embedding of the pair attacked in words space. Under the MoCo setting, this is illustrated in the figure 2.9. The algorithm of the MoCo implementation can be found in the annex.

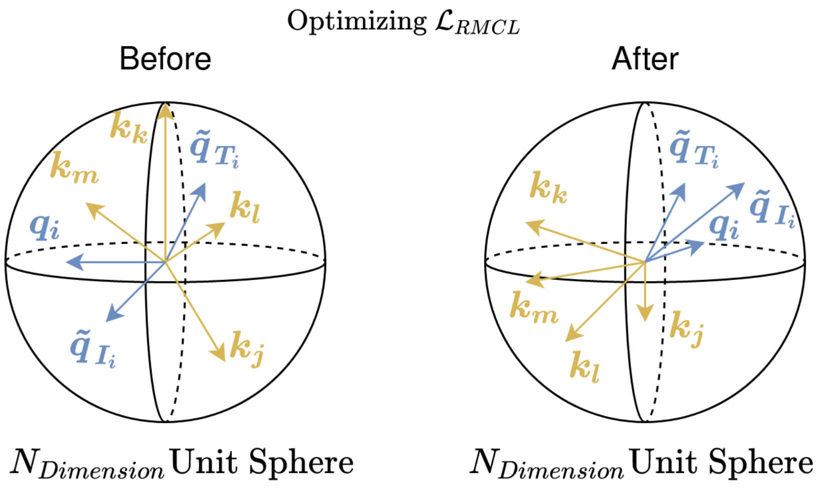

Optimizing $\mathcal{L}_{RMCL}$
Before
After
$\boldsymbol{k_k}$
$\tilde{\boldsymbol{q}}_{\boldsymbol{T_i}}$
$\boldsymbol{k_m}$
$\boldsymbol{k_l}$
$\boldsymbol{q_i}$
$\tilde{\boldsymbol{q}}_{\boldsymbol{I_i}}$
$\boldsymbol{k_j}$
$\tilde{\boldsymbol{q}}_{\boldsymbol{T_i}}$
$\boldsymbol{k_k}$
$\tilde{\boldsymbol{q}}_{\boldsymbol{I_i}}$
$\boldsymbol{q_i}$
$\boldsymbol{k_m}$
$\boldsymbol{k_j}$
$\boldsymbol{k_l}$
$N_{Dimension}$ Unit Sphere
$N_{Dimension}$ Unit Sphere

Figure 2.9: Geometrical representation of the optimization of the $L_{RMinfoNCE}$. The projected representation $\boldsymbol{q_i}$ and $\boldsymbol{k_i}$ are 128-dimensional vectors L2 normalized. By optimizing the loss we are pulling the attacked joint-representation $\tilde{\boldsymbol{q_{I_i}}}$ and $\tilde{\boldsymbol{q_{T_i}}}$ closer the original joint-representation vector $\tilde{\boldsymbol{q_i}}$ and pushing apart all the dissimilar joint-representation vector $\boldsymbol{k_{j \neq i}}$. Note that the in BarlowTwins the vectors are not L2 normalized as discuss in section 2.4

Similarly, the Robust Multimodal BarlowTwins loss is defined as follow :

$$
\begin{aligned}
\mathcal{L}_{\mathcal{RMBT}} &\triangleq \sum_{\mathcal{C}\in\{\mathcal{C}^{\mathcal{I}},\mathcal{C}^{\mathcal{T}}\}} \left[\underbrace{\sum_i (1-\mathcal{C}_{ii})^2}_{\text{invariance term}} + \lambda \underbrace{\sum_i \sum_{j\neq i} \mathcal{C}_{ij}^2}_{\text{redundancy reduction term}} \right] \\
\mathcal{C}_{ij}^{I} &\triangleq \frac{\sum_b \tilde{\boldsymbol{q}}_{\boldsymbol{b},\boldsymbol{i}}^{\boldsymbol{I}} \boldsymbol{k}_{\boldsymbol{b},\boldsymbol{j}}}{\sqrt{\sum_b \left(\tilde{\boldsymbol{q}}_{\boldsymbol{b},\boldsymbol{i}}^{\boldsymbol{I}}\right)^2} \sqrt{\sum_b \left(\boldsymbol{k}_{\boldsymbol{b},\boldsymbol{j}}\right)^2}} \\
\mathcal{C}_{ij}^{T} &\triangleq \frac{\sum_b \tilde{\boldsymbol{q}}_{\boldsymbol{b},\boldsymbol{i}}^{\boldsymbol{T}} \boldsymbol{k}_{\boldsymbol{b},\boldsymbol{j}}}{\sqrt{\sum_b \left(\tilde{\boldsymbol{q}}_{\boldsymbol{b},\boldsymbol{i}}^{\boldsymbol{T}}\right)^2} \sqrt{\sum_b \left(\boldsymbol{k}_{\boldsymbol{b},\boldsymbol{j}}\right)^2}}
\end{aligned}
\tag{2.10}
$$

2.5 Augmentation

Initially, data augmentation was a hack to make the supervised model more robust. In contrastive learning, it's the central ingredient. Data augmentation might serve as a proxy for what happened in real life. Indeed, Humans are like agents navigating in the words observing objects in various circumstances, and different transformations have been applied to their viewpoint. The data augmentation used in contrastive learning is different compared to the one used in self-supervised learning. For instance, aggressive color distortion is essential in computer vision contrastive methods (Chen et al., 2020 [12]). The intrinsic setting of contrastive methods may exploit the shortcut of overfitting on the color histogram instead of learning the richer information. The contrastive methods compare an augmented view of the same image. Indeed, not all dogs have the same color histogram, but in contrastive learning, all dogs' individual images will have the same color histogram. Consequently, the data augmentation in self-supervised learning is carefully designated to maintain their instance identities so that the transformed sample from the same instance can still be retrieved.

Although existing contrastive learning literature discussed their boosts on the standard generalization (Dosovitskiy et al., 2014 [49]; Oord et al., 2018 [33]; Wu et al., 2018 [34]), many others attest that the feature consistency is valuable for robustness too (Ziyu et al., [50]; Kim et al. 2020 [21]). One interpretation of adversarial brittleness could be related to the non-smooth feature space near instances. Indeed, slight input perturbations can produce notable feature variations and may even change the labels. Reinforcing agreement during training w.t.r perturbations has been therefore attested to help adversarial robustness instantly. Indeed, Kim et al. 2020 [21] have shown that using adversarial perturbation to create “hard" positives in the contrastive loss is effective to enhance the robustness. Ziyu et al., [50] further investigate the performance of robust contrastive learning by analyzing three candidate algorithms. Their first option (A2A) was to attack the two branches and compute the similarity on different adversaries of the same image. Their second option (A2S) was to attack only one branch and compute the similarity between an adversary and an augmented view from the same images. Finally, their third option (DS) was to use both approaches simultaneosly. They empirically show that A2A was overly disruptive and only degrading the feature quality. Indeed, they point out that a standard SimCLR architecture improves in terms of robustness while keeping similar performance in terms of accuracy when trained in A2S and DS settings.

The above considerations compose our hypothesis that multi-modal contrastive learning could be an excellent option for adversarial pre-training. In this work, we will use adversaries as augmented views in our multi-modal contrastive learning framework. To be specific, we will compute the similarity between an attacked pair and a clean pair. We will compare the performance of this attacked view with regular augmented views. We refer to Robust multi-modal contrastive learning (RMCL) when using adversaries and multi-modal contrastive learning (MCL) when using standard data augmentation.

In the following subsection, we will present the types of images and text augmentation used with our baseline MCL (subsection 2.5.1) and the adversaries for our RMCL methods.(subsection 2.5.2)

2.5.1 The clean views

Images : RandAugment

SimCLR proposes multiple forms of data augmentation for images such as crop and resize, rotate, Gaussian blur, color jittering, and many other image views. These augmentations have been carefully selected to suits any contrastive approach. Indeed, BYOL and BarlowTwins use the same augmentation parameters for image augmentations. Inspired by these methods, we will use the same type of image augmentation.

Text : EDA and PEGASUS

It is more challenging to construct text augmentation, which does not alter the semantics of a sentence. Indeed, a single token can invert the meaning of a sentence. Fang et al. 2020 [51] improve the performance of BERT on various downstream understanding tasks by using contrastive learning with back-translation (Sennrich al., 2015 [52]) and easy-data-augmentation (EDA ;Wei et al., 2019[53]). In this work, we will use EDA (Easy Data Augmentation; Wei & Zou 2019) and PEGASUS (Zhang et al., [54]) as augmented strategies for text. Initially, PEGASUS is a sequence-to-sequence pre-training objective tailored for abstractive text summarization. However, it has shown great performance when fine-tuning for paraphrasing.

Specifically, each sentence will be augmented five times with EDA, and five times with PEGASUS then we will rank them based on their semantic similarity with the original sentences. To achieve this, we compute for every augmented sentence it's sentence features vector with SentenceBert (Reimers et al., 2019 [55]) and compute a similarity score by computing the cosine similarity between the original sentence vector, and it's augmented versions. Once the ranking is established, we select a different augmented view for each training epoch in descending order.

2.5.2 The attacked views

In RMCL we use adversarial training as an adequate regularization to enhance model generalization. It is achives by minimizing the following objective :

$$\min_{\boldsymbol{\theta}} \mathbb{E}_{(\{I_i,T_i\})\sim\mathcal{D}} \left[\max_{\boldsymbol{\delta}_{img}\in\mathcal{S}} L_{RMCL} \left(g(f_{\boldsymbol{\theta}} \left(\{I_i + \boldsymbol{\delta}_{img}, T_i\} \right)) \right) \right] \tag{2.11}$$

where $\{I_i, T_i\}$ is an image text pair from the dataset $\mathcal{D}$, g_θ is the linear projector, f_θ is the encoder, $\boldsymbol{\delta}_{img}$ is an image perturbation from a set $\mathcal{S}$ of allowed perturbations. It is a label-free reformulation of the saddle point problem in adversarial training presented by Madry et al., 2017 [23]. It is composed of an inner maximization problem and an outer minimization problem. The inner maximization problem aims to obtain an adversarial variant of a given pair $\{I_i, T_i\}$ that maximizes the loss. On the other hand, the outer minimization problem attempts to obtain the model parameters that minimize the "adversarial loss" produced by the inner attack problem. When pre-training V&L architectures, the location embeddings are employed to encode the location of image patches and sub-word tokens. Our adversaries alter the image's pixel space and the word's space, letting the rest of the elements fixed. Moreover, because of the diverse aspects of image and text modalities, we suggest attacking one modality at a time.

Images : Projected Gradient Descent

Madry et al., 2017 [23] have shown that for images, the inner maximization problem in equation 2.11 can be solved accurately by PGD, a conventional method for constrained optimization. In their works, they demonstrate that a PGD-based attack is the ultimate first-order adversary for images. Robustness against the PGD adversary yields robustness against all first-order adversaries. Inspired by VILLA (Gan et al., [56]), which uses PGD based attacks to improve their V&L model robustness, we will also use PGD attacks to generate our image view adversary. However, our work differs from VILLA since they use a PGD-based attack on the features space while we use PGD in the input space (Pixel space).

To be specific, let us take a pixel-wise perturbation $\boldsymbol{\delta}_{img}$, PGD do the following step (with step-size α) in each iteration:

$$\boldsymbol{\delta}_{img,t+1} = \Pi_{\|\boldsymbol{\delta}_{img}\| \leq \epsilon} \left(\boldsymbol{\delta}_{img,t} + \alpha g\left(\boldsymbol{\delta}_{img,t}\right) / \left\| g\left(\boldsymbol{\delta}_{img,t}\right)\right\|_2 \right) \tag{2.12}$$

where $g = \nabla_{\boldsymbol{\delta}_{img}} L_{RMCL}\left(g(f_{\boldsymbol{\theta}}\left(\{I_i + \boldsymbol{\delta}_{img}, T_i\}\right))\right)$ is the gradient ot the loss w.r.t. $\boldsymbol{\delta}_{img}$, and $\Pi_{\|\boldsymbol{\delta}_{img}\| \leq \epsilon}$ makes a projection onto the $\epsilon-$ball. After the PGD attack $\boldsymbol{q_i}$ becomes $\tilde{\boldsymbol{q}}_{\boldsymbol{i}}^{\boldsymbol{I}}$

Text : Geometric inspired attack

In NLP, adversarial training in the input space has been challenging, as existing natural language adversarial attacks are too slow to generate adversarial examples on the fly during training (Alzantot et al., 2018 [57]; Ren et al., 2019 [58]). Meng et al., 2020 [24] has proposed a solution to this issue by presenting a geometry-inspired attack for generating natural language adversarial examples. Furthermore, Meng et al., 2021 [22], has shown the effectiveness of this method through the improvement of the pre-trained language model BERT by leveraging contrastive learning with their attack. The geometric-inspired attack is an efficient adversarial attack that enables word-level adversarial training. Inspired by their methods that improve robustness without labels, we also use the Geometry attack to generate natural language adversarial examples.

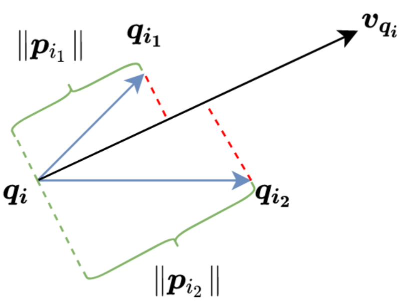


Figure 2.10: An illustration of one iteration in Geometry Attack for multimodal contrastive loss. Refer to section 2.5.2 for more details.

The intuition behind the attack is to iteratively replace words in the original texts such that in each repetition the replaced word increases the contrastive loss as much as possible. To be specific, consider a text example T_i with its corresponding images I_i,their joint embedding $\boldsymbol{q_i}$ and and $L_{RMCL}\left(g(f_{\boldsymbol{\theta}}\left(\{I_i, T_i\}\right))\right)$ as ℓ_i, we then have :

1. Compute the gradients of ℓ_i with respect to $\boldsymbol{q_i}$. It allows us to know in which direction we should move the joint embedding $\boldsymbol{q_i}$ to increase the contrastive loss ℓ_i. We have the gradient vector $\boldsymbol{v}_{q_i} = \nabla_{\boldsymbol{q}_i}\ell_i$

2. Solve the gradients of ℓ_i w.r.t input word embeddings of T_i. This steps let us take advantage of the gradient of each tokens of the tokenized text sample T_i. Therefore, we can understand which word has the most influence in the computation of ℓ_i. The words w_t are then ranked in descending order based on their gradient score.

3. We pick the words w_t with the highest impact on the ℓ_i and we derive a synonym set of maximum M element $\mathbb{Q}_t = \left\{w_k^0, w_k^1, \ldots, w_k^M\right\}$.The candidate set of synonym is initiated with N closest synonyms according to the cosine similarity between w_t and every other word in the vocabulary . Following Jin et al. 2020 [59], we filter out semantically different words from the candidate set by discarding candidate words of which the cosine similarity of their embeddings between the embeddings of w_t below a threshold ϵ.

4. Once we have the synonym set of the given selected word w_t we compute the joint-embeddings vectors $\left\{\tilde{\boldsymbol{q}}_{i_1}, \tilde{\boldsymbol{q}}_{i_2}, \cdots, \tilde{\boldsymbol{q}}_{i_M}\right\}$. We then compute the delta vector $\boldsymbol{r}_{i_j} = \tilde{\boldsymbol{q}}_{i_j} - \boldsymbol{q}_i$. The projection of $\boldsymbol{r}_{i_j}$ onto $\boldsymbol{v}_{q_i}$ is $\boldsymbol{p}_{i_j} = \frac{\boldsymbol{r}_{i_j} \cdot \boldsymbol{v}_{q_i}}{\|\boldsymbol{q}_{z_i}\|}$. We select the candidate word w_{t_m} in T_i where $m = \text{argmax}_j \left\|\boldsymbol{p}_{i_j}\right\|$. In other words, w_{t_m} results in the largest projection $\boldsymbol{p}_{i_j}$ onto $\boldsymbol{v}_{q_i}$

5. Finally we replace w_t with w_{t_m} in the sentence T_i and we have $\boldsymbol{q}_i \leftarrow \tilde{\boldsymbol{q}}_{i_m}$. We repeat step 1-4 N iteratoin, where N is a hyperparameter of the method. We expect ℓ_i to increase in each iteration.

Figure 2.10 illustrates an iteration of our attack. This attack can be easily implemented in a batched fashion, making it possible to generate adversarial examples on the fly during training. After the geometric-inspire attack $\boldsymbol{q_i}$ becomes $\tilde{\boldsymbol{q}}_{\boldsymbol{i}}^{\boldsymbol{T}}$

Chapter 3
Methodology

This chapter describes our methodology for pre-training V&L models with our Robust Multimodal Contrastive Learning task. The rest of this chapter is organized as follows: Section 3.1 presents the pre-trained ViLT architecture and the parameter of our MoCo and BarlowTwins RMCL frameworks. Furthermore, we review the parameter of the augmentation and the adversaries. Afterward, We delineate the pre-training tasks in section 3.2 as well as the downstream methods in section 3.3. Finally we explicit the implementation details in section 3.4

3.1 The models Architecture

A diagram of our RMCL framework is given in Figure 2.8. We use the pre-trained ViLT-B/32 (Kim et al., 2021 [4]) as encoders. It takes a concatenation of image and text inputs as depicted in figure 2.7. To be more specific, ViLT uses a 32 ×32 patch projection inspired by ViT (Dosovitskiy et al., 2020 [25]) as a visual embedder and the bert-base-uncased tokenizer to tokenize the text inputs. The concatenated sequence of visual and textual features are fed in the ViLT encoder, and the encoded sequence is pooled to get a single vector $\boldsymbol{CLS}$ that represents the pair. The joint representation $\boldsymbol{CLS}$ is a 768 dimension vector. The $\boldsymbol{CLS}$ vector is then projected by either the MoCo head either the BarlowTwins head and the $\boldsymbol{q}$ and $\boldsymbol{k}$ embedding vectors resulting are used in their respective contrastive loss (equation 2.9 and equation 2.10). The BarlowTwins projector network has three linear layers, each with 8192 output units, while the MoCo projector has two linear layers with respectively 768 and 128 output units. As we discuss in section 2.2 the infoNCE is prone to the curse of dimensionality, implying a low-dimensional setting. In the MoCo mechanism, the negative keys $\boldsymbol{k}_{-}$ are maintained in a queue, and only the queries $\boldsymbol{q}$ and positive keys $\boldsymbol{k}_{+}$ are encoded in each training batch. Following Chen et al., 2020 [3], we use a queue size of 65536, a momentum update parameter of 0.999 and the default parameter $\tau = 0.07$. Following Zbontar et al., 2021 [2] we use a trade-off hyperparameter $\lambda = 5 \cdot 10^{-3}$ for BarlowTwins loss function. For our baseline MCL we use the same image augmentation than Zbontar et al., 2021 [2] while we use PEGASUS and EDA for text augmentation (see discussion in section 2.2). In the Robust MCL settings, we use PGD as attacked on the pixel space and the geometric-inspired attack in words space. The hyperparameter of PGD (equation 2.12) are the $\epsilon-$Ball and attacked rate α. The hyperparameter of the Geometric-inspired attack is the number of synonym candidates M and the number of iterations max_{loop}. We will conduct in chapter 4 a study to select the best adversarial hyper-parameters. For all experience, we will keep the original optimizer of ViLT (Kim et al., 2021 [4]).

3.2 The pre-training tasks

RMCL and MCL are both pre-trained with either MoCo or BarlowTwins settings with the same image-text datasets. Initially we wanted to use four datasets with our pre-training task: Microsoft COCO (Lin et al., 2014 [60]), Visual Genome (VG) (Krishna et al., 2017 [61]), SBU Captions (SBU) (Ordonez et al., 2011 [62]), and Google Conceptual Captions (GCC) (Sharma et al., [63]). Table 1 reports the dataset statistics. However, due to time constraints, we only pre-trained on COCO for a single epoch instead of 10 as intended.

Dataset	#Images	#Captions	Caption Length
Conceptual Caption	2,68M	2,68M	10.66 ± 4.93
SBU	780K	780K	15.0 ± 7.74
COCO	82K	414K	11.81 ± 2.81
Visual Genome	86K	4,32M	5.53 ± 1.76
Total	3,63M	8,20M	

Table 3.1: Statistics of pre-training datasets.Caption length is the length of tokens from pre-trained bert-base-uncased tokenizer.

3.3 The downstream tasks

We evaluate the robustness and accuracy performance of our RMCL task on ViLT with two generally explored types of vision-and-language clssification tasks: NLVR2 (Suhr et al., 2018 [26]) and VQAv2 (Goyal et al., 2017 [27]). Futhermore we evaluate the retrieval image and text tasks with COCO and Flickr30K (F30K; Karpathy et al., 2015 [28]). The table 3.2 shows statistics about the downstream tasks. It is worth mentioning that NLVR2 and Flickr30K are out-of-domain datasets, while COCO and VQAv2 are in-domain. Indeed, with out-of-domain tasks, the dataset is not the same as for pre-training. For the classification and retrieval tasks, we follow the original ViLT (Kim et al., 2021 [4]) by fine-tuning for ten epochs with a batch size 256 for retrieval tasks and VQAv2 and 128 for NLVR2. The statistic of these downstream tasks are displayed in table 3.2.

While we process the evaluation on the downstream tasks, we gauge the robustness of ViLT-pre-trained with RMCL by attacking the images and text inputs independently. We study the robustness when either attacking a single modality or either attacking both modalities independently. We compare the robustness and accuracy performance of ViLT with and without being pre-trained with RMCL.

Task	Image Source	#Images	#Captions
VQAv2	COCO	204K	1.1M
NLVR2	Web Crawled	214K	107K
Image-Text Retrieval	COCO	92K	460K
	Flickr30K	32K	160K

Table 3.2: Statistics on the datasets of downstream tasks

3.4 The implementation Details

Following ViLT (Kim et al., 2021 [4]), we use for the classification tasks a downstream head composed of two-layer MLP of hidden size 2*768. Furthermore, for the retrieval tasks, we initialize the similarity score from a fine-tuned ITM head. The ITM head is a single linear layer that projects the pooled output feature ***CLS*** of size 768 to a binary class. We finetuned VQA, NLVR2 and IRTR with Flickr30k on 10 epochs and IRTR with COCO on 2 epochs. Indeed, the fine-tuning of IRTR with COCO is computationally intence and due to our time constrain we reduce the number of epochs.

Question Answering VQAv2 uses pairs of an image and a question and asks for answers. A usual practice is to transform the problem into a classification task with 3,129 answer classes (Yu et al., 2019 [64])). We fine-tune on VQAv2 for ten epochs with a batch size of 256 using a binary cross-entropy loss.

Natural Language for Visual Reasoning The goal in NLVR2 is to decide whether a natural language description is true w.r.t the given image pair. Unlike the pre-training setting there are two input image. Various strategies have been suggested.[1]. We will use the pair methods following ViLT (Kim et al., 2021 [4]), OSCAR (Li et al,. 2020 [65]) and VinVL (Zhang et al., 2021 [66]). In the Pair setup, the triplet input is changed to a two pairs (Image1, Question1) and (Image2, Question1) formulation. Each pairs are then fed into the encoder. The

[1] UNITER (Chen et al., 2019 [41]) presented three settings: pair, triplet, and pair-biattn.

NLVR2 projector takes the concatenation of the two pooled joint representations $\boldsymbol{CLS}$ as input and outputs a binary prediction.

Retrieval Tasks Retrieval tasks use as well the $\boldsymbol{CLS}$ joint representation. In V&L, the task consists of text and image retrieval. For example, image retrieval consists of identifying an image from a collection of sentences describing it. It is similar to text retrieval, where the task is to identify a text from a pool of images. Due to time-constrained, we report only the evaluation performance without attacks. Indeed, the evaluation on IRTR takes up to 1 day without attacks. The correlation score is achieved through the fine-tuned of the ITM head, especially the section that calculates the true-pair logits. Fifteen randomly selected negative texts are used, and the model is tuned with cross-entropy loss that maximizes the scores on the true image caption.

Computation Our models are implemented based on PyTorch lightning [2]. We used several PyTorch lightning plugins to speed up training, such as mixed precision (Micikevicius et al,. 2017 [67]), Distributed data-parallel (Li et al,. 2020 [68]), and a data loader with several workers. All our models have been pretrained on 8 GeForce RTX 3090 and fine-tune on 2 GeForce RTX 3090. Gradient accumulation (Ott et al., 2018 [69]) is also applied to reduce multi-GPU communication overheads.

[2] https://www.pytorchlightning.ai/

Chapter 4

Results and Discussion

In this section, we present and discuss the various experience and their results. We start by conduction in section 4.1 a hyper-parameter selection for the attacks by attacking the pre-trained ViLT-B/32 model while doing an evaluation on the downstream task NLVR2. We select some geometric-attacked samples and observe the semantic conservation. Once the parameter for each attack is selected, we study the characteristic of the RMCL pre-training process on ViLT in section 4.2. Furthermore, The evolution with respect to the optimizer steps of the loss, the distances between positive and negative pairs as well as the attacks characteristic are reviewed. Finally, in the section 4.3 we present the robustness with NLVR2 and the image and text retrieval performance under different settings.

4.1 Adversaries HyperParameter Selection

In this section, we conduct an adversaries hyperparameter selection by attacking the evaluation process of ViLT on NLVR2. To be specific, we use the ViLT-B/32 finetuned on NLVR2 for ten epochs and attack the binary-cross-entropy loss of the tasks during evaluation. The original accuracy on the task is 74.48% without attack.

4.1.1 Image: Projected Gradient Descent

The PGD ensure imperceptibility of the perturbation $\boldsymbol{\delta}_{img}$ limiting its l_∞-norm ϵ. A larger ϵ makes the attack stronger, resulting in a higher error rate of the model, but also makes the perturbation more perceptible to the human. In order to have a comparison, we illustrate in Figure 4.1 the distribution of the average norm of the row pixel from the NLVR2's images. Then, we report the NLVR2 accuracy and success rate [1] for a range of $\epsilon-$Ball $\{1e^{-2}, 8e^{-3}, 5e^{-3}, 3e^{-3}, 2e^{-3}, 1e^{-3}, 5e^{-4}\}$[2]

[1] Sucess rate refers to the change rate of the algorithm prediction when attacked versus when not attacked

[2] Much more parameters have been tested. We do not plot them for rendering purposes

and attacked rate α $\{8e^{-2}, 5e^{-2}, 2e^{-2}, 8e^{-3}, 5e^{-3}, 1e^{-3}\}$ with a fixed number of 5 PGD's iterations.

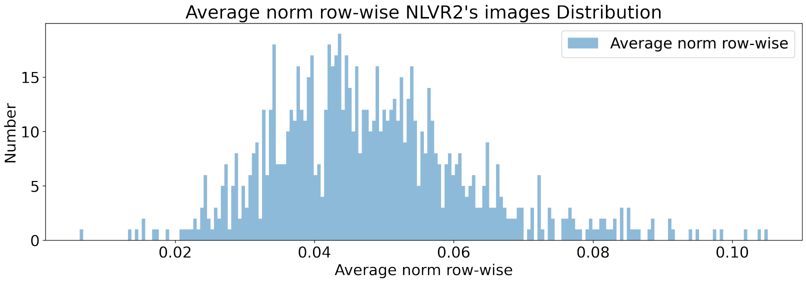


Figure 4.1: Distribution of the average norm of the row pixel from the NLVR2's images

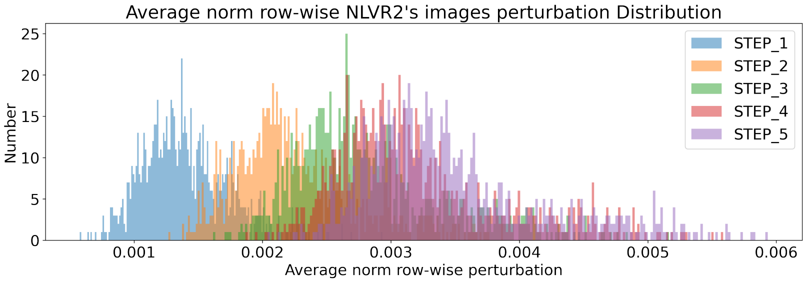


Figure 4.2: Distribution of the average norm of the row pixel from the image perturbation $\boldsymbol{\delta}_{img}$ for the parameters : $\alpha = 0.05$ and $\epsilon = 0.005$. The steps correspond to the iteration of the PGD. The average norm perturbation is an order of magnitude smaller than the average norm of the images.

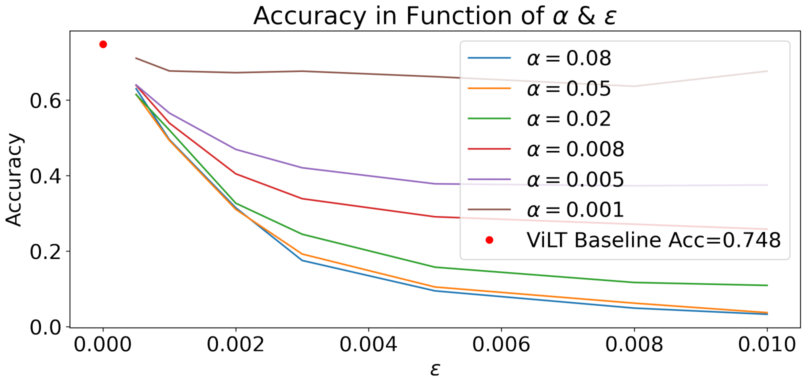


Figure 4.3: NLVR2 test-accuracy in function of PGD's hyper-parameters

Figures 4.3 4.4 4.6 depict the expected behaviors of the hyperparameters. Indeed, when the $\epsilon-$Ball is more restrictive, the model accuracy is higher, the

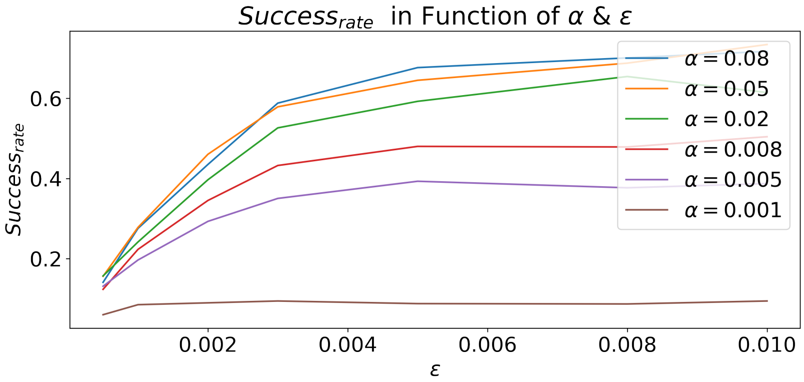


Figure 4.4: Success rate of the PGD attack on the NLVR2 accuracy in function of PGD's hyper-parameters

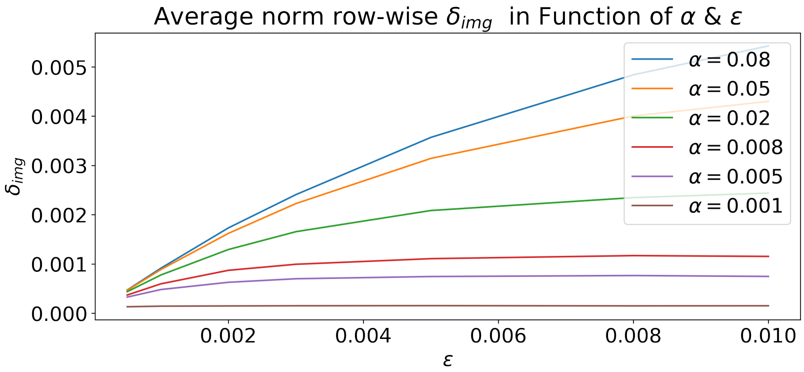
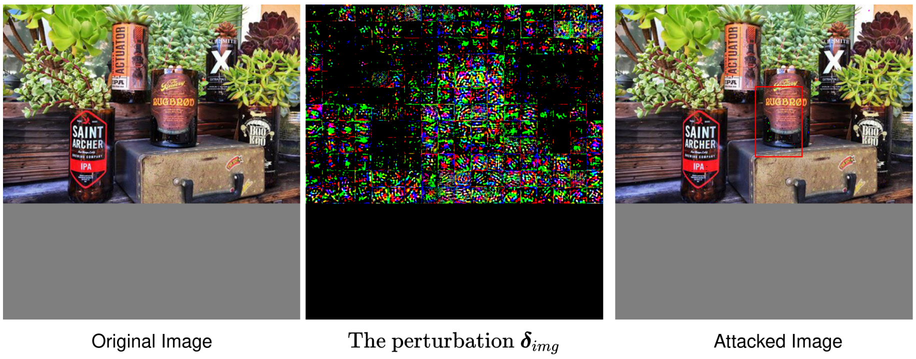


Figure 4.5: Average norm of $\boldsymbol{\delta}_{img}$ from the PGD attack on the NLVR2 in function of PGD's hyper-parameters

success rate is lower and obviously the norm of the perturbation $\boldsymbol{\delta}_{img}$ is smaller. Similarly, when the parameter α is higher, the model accuracy drops, the success rate increased and the $\boldsymbol{\delta}_{img}$ perturbation gets more aggressive.

Regarding the result in Figure 4.3-4.6 $\epsilon = 0.005$ and $\alpha = 0.05$, seem to be fair enough choices. Indeed, the algorithm is strongly impacted in terms of accuracy with a drop of 65% while having perturbation of an order of magnitude smaller than the original images (Figure 4.3-4.2).To be specific, our choice of PGD parameters leads to an accuracy of 10.4%, a success rate of 64.5%, and an average norm perturbation of 0.003. We display an example of the clean image, the perturbation added to the image and the resulting attacked images in figure 4.6. The attack have successfully change the model prediction while being almost imperceptible. We can note that the attack is dense, and almost every pixel seems to be attacked. Nevertheless, on the attacked image on the left, we can only identify very few disturbed pixels highlighted with a red box. it is also fascinating to observe that the attacks reveals the 16x16 image patches



Original Image

The perturbation $\boldsymbol{\delta}_{img}$

Attacked Image

Figure 4.6: An example of adversarial attacks using projected gradient descent. PGD results in dense perturbations, in which almost every pixel is perturbed. It is fascinating to note that we can distinguish in the perturbation the different patches of the images. We add on the left image a red box that highlights some visible perturbation.

4.1.2 Text: Geometric inspired attack

The geometric-inspired attack fools natural language models with high success rates while only replacing a few words. By design, the attack ensures a words-replacement strategy that aims to keep the semantics meaning of the sentence. We report the NLVR2 accuracy (Figure 4.7) and success rate (Figure 4.8) for a range of number of synonym candidate M $\{10, 8, 5, 2, 1\}$ and maximum number of iteration max_{loop} $\{10, 8, 5, 2, 1\}$. Furthermore, we illustrate in Figure 4.9 the rate of words changed per sentence in the function of the aforementioned list of hyper-parameters. Outside the performance metrics, the computation time and memory requirement need to be taken into account. We haven't explicitly measured these metrics however our experiments have shown that above a maximum number of candidate of 10 the algorithm requires a prohibitive amount of CPU-memory.

Figure 4.8 depicts the expected behaviors of the hyperparameters. Indeed, when the number of synonym candidates M is higher, the success rate of the attack is higher. In fact, the synonyms will have a lower similarity score with the original word that may induce a higher mismatch with the original pair, thus a higher gradient of the contrastive loss. Furthermore, when the number of iteration gets higher, the loss is expected to increase (see discussion in section 4.1). Based on the results and the memory issue, we have selected a number of synonyms $M = 5$ and $max_{loop} = 10$. Indeed, the choice of the parameters ensures a high success rate $\sim 48\%$ with a low rate of words changed $\sim 20\%$ while not requiring excessive CPU-memory usage. Indeed, the cosine similarity matrix

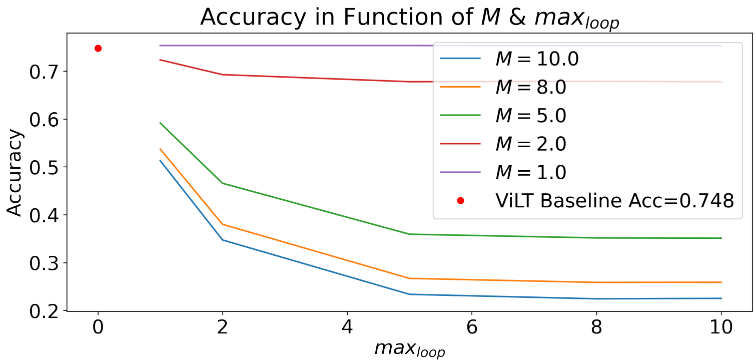


Figure 4.7: NLVR2 test-accuracy in function of Geometric inspired attack's hyper-parameters

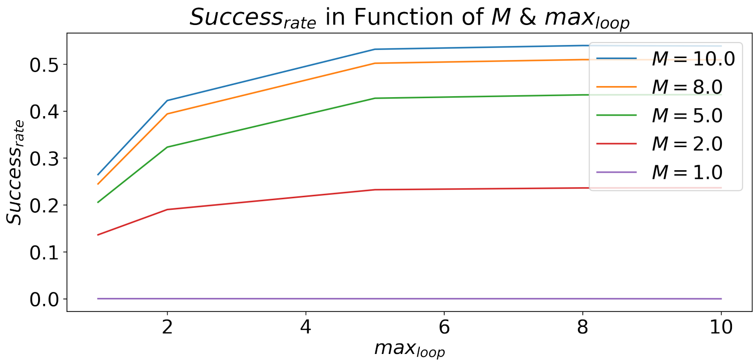


Figure 4.8: Rate of words changed in a sentences for the Geometric attack on NLVR2 for various hyper-parameter

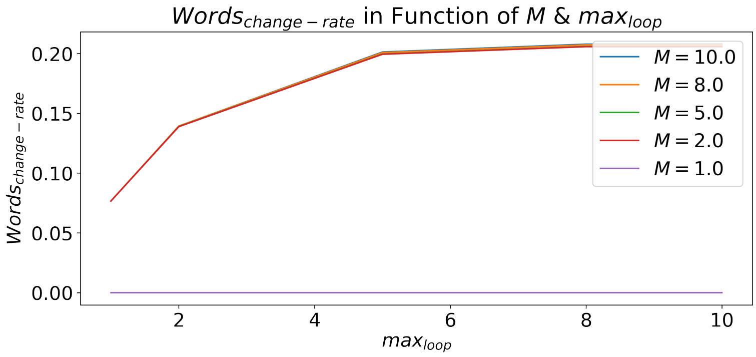
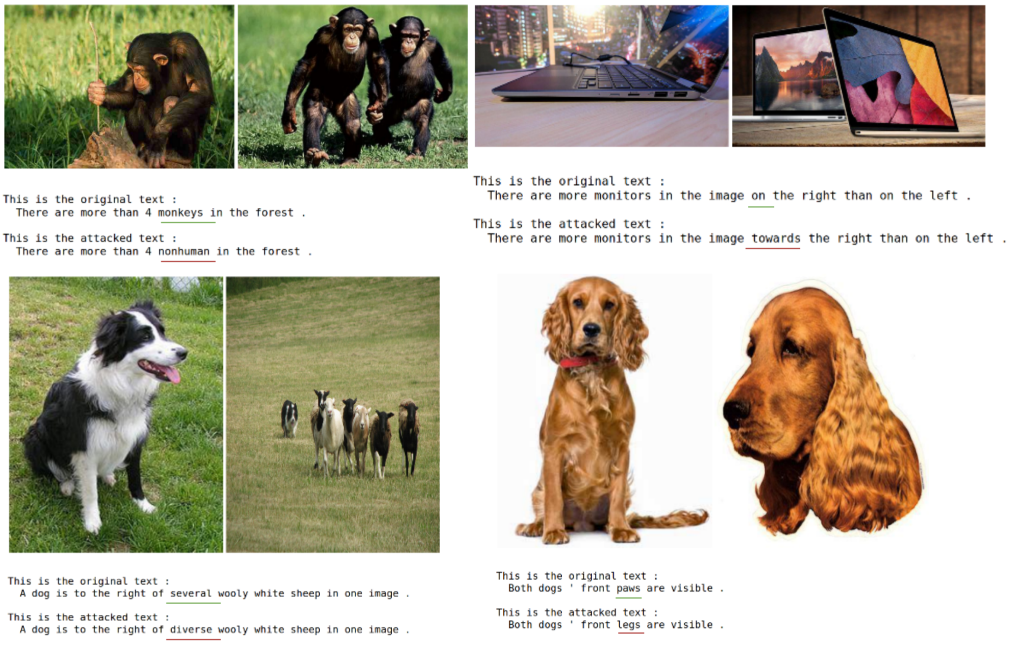


Figure 4.9: Success rate of the Geometric attack on NLVR2 for various hyper-parameter

between words is computed in the function of the max number of synonym M.

Some examples of the attack with the chosen parameter are illustrated in Figure 4.10.



```
This is the original text :
  There are more than 4 monkeys in the forest .

This is the attacked text :
  There are more than 4 nonhuman in the forest .
```

```
This is the original text :
  There are more monitors in the image on the right than on the left .

This is the attacked text :
  There are more monitors in the image towards the right than on the left .
```

```
This is the original text :
  A dog is to the right of several wooly white sheep in one image .

This is the attacked text :
  A dog is to the right of diverse wooly white sheep in one image .
```

```
This is the original text :
  Both dogs ' front paws are visible .

This is the attacked text :
  Both dogs ' front legs are visible .
```

Figure 4.10: Example of image-text pairs attacked in the word-space with the Geometric-inspired. In each example, the attack has successfully changed the prediction. The words in green are the original, and the ones in red the adversary changed. Parameter attacks : $M = 5$ and $max_{loop} = 10$

4.2 pre-training RMCL BarlowTwins and MoCo

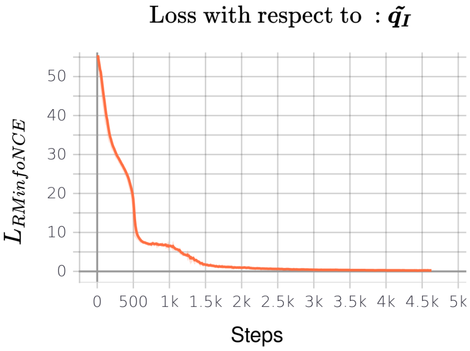
Once the parameters for the Geometric-inspired and the PGD attacks are selected, we conduct a full RMCL pre-training task with MoCo and BarlowTwins settings on ViLT-B/32 and COCO datasets. We run the pre-training task for one epoch on a single 8 GPUs node with a batch size of 128 for 1 days for each setup. Originally, we planned to run the pre-training on SBU-COCO-VG-CC for 10 epoch however, due to time constrain, we reduce the pre-training to 1 epochs on only COCO. The following results serve as a proof of concept.

In the following, we study the losses evolution for both methods in subsection 4.2.1. We verify in subsection 4.2.2 that the distance between positive pairs is decreasing while it's increasing between positive and negative pairs. Finally, we briefly look at some adversaries metrics in the subsection. 4.2.3.

4.2.1 Loss

MoCo

The Figure 4.11 represent the evolution of the two-term of the robust multimodal MoCo loss from the equation 2.9. We remark that the loss w.r.t $\tilde{q_I}$ and the loss w.r.t $\tilde{q_T}$ decreased significantly although attacked. Furthermore, we observe that both losses are very similar. These observations demonstrate that our algorithm is learning from our task.



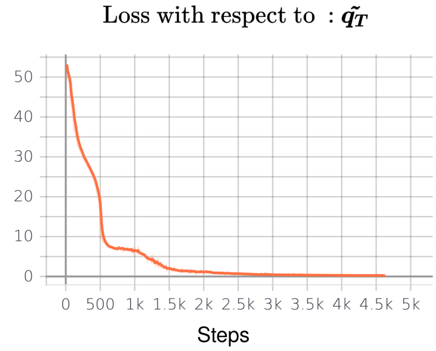


Figure 4.11: Robust multimodal InfoNCE loss evolution during 1 epochs on COCO dataset. The two component of the loss in equation 2.9 are illustrated.

BarlowTwins

Figure 4.12 depicts the evolution of the four terms of the robust multimodal BarlowTwins loss from the equation 2.10. We observe that both invariance term for $\tilde{q_I}$ and $\tilde{q_T}$ decreased significantly and reach after $\sim$ 4k steps a loss of almost 500. However, we notice that the initial invariance loss w.r.t $\tilde{q_T}$ is higher than the initial loss invariance w.r.t $\tilde{q_I}$. It implies that the geometric attack have a slightly higher impact on the BarlowTwins loss than the PGD. Futhermore, the invariance loss plots haven't meet yet any characteristic of a steady state as opposed to the MoCo losses Figure 4.11 . It suggest that BarlowTwins may benefits more from an increases of epoch than MoCo on COCO dataset only. The same study hold for the two redundancy reduction term. However, we discern that the redundancy losses are much higher than the invariance losses and reach a plateau after 3.5k steps. From these observations, we conclude that our algorithm is learning from our task despite being attacked.

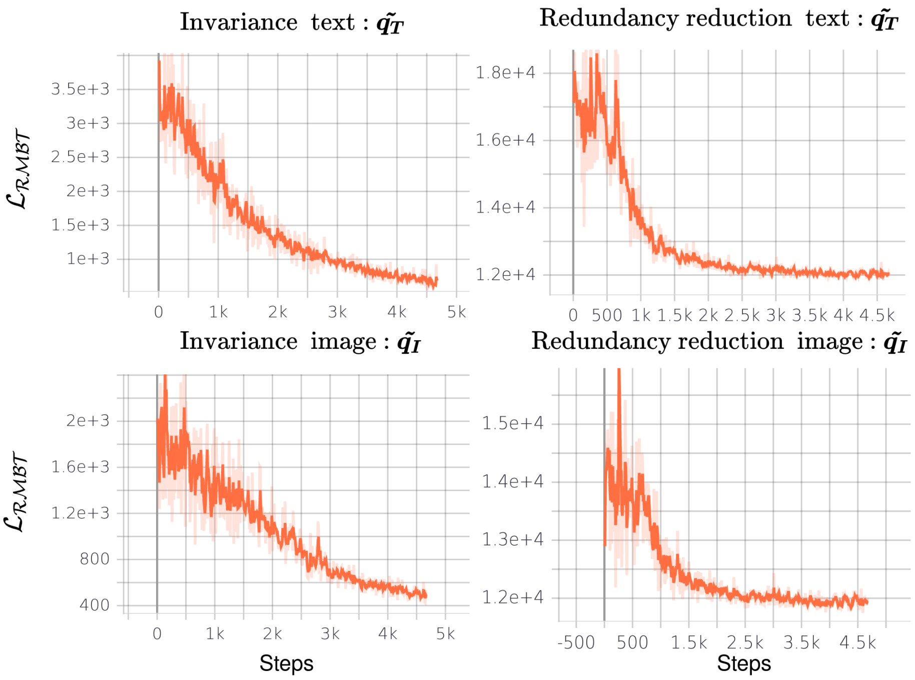


Figure 4.12: Robust multimodal BarlowTwins loss evolution during 1 epoch on COCO dataset. The four component of the loss in equation 2.10 are illustrate.

4.2.2 Distances

In this subsection, we depict the similarity relation between the positive and the negative joint-embedding. We use the standard L_2 euclidean-distance as well as the cosine similarity angle. The cosine similarity has the advantage over the L_2 euclidean-distance of being more adapted for high dimensional vectors. Indeed, Moco has 128 dimensions L_2-normalized embedding vectors while BarlowTwins have 8192 dimension non-normalized embedding vectors. (see discussion in section 2.2.4).

MoCo

The Figures 4.13 and 4.14 delineate the evolution of respectively the positive and negative L_2 and $cosine_{sim}$ distances during the pre-training with the robust multimodal Moco framework. We observe that all the graphs from both figures 4.13-4.14 overshoot within the two first thousand steps. This is due to the random initialization of the Moco memory bank. We can argue that once the queue is filled with batches of negative pairs $\boldsymbol{k}_-$ the algorithm starts its learning process. Indeed, this effect can also be seen in the two first thousand steps of the losses in figure 4.11. Apart from this observation, we see that our task is inducing the expected behavior discuss in figure 2.9. The distance between positive sample (respectively $\tilde{\boldsymbol{q}}_{\boldsymbol{I}}$ with $\boldsymbol{k}_+$ and $\tilde{\boldsymbol{q}}_{\boldsymbol{T}}$ with $\boldsymbol{k}_+$ are decreasing w.r.t L_2 euclidean-distance and converging to 1 w.r.t cosine similarity (Figure 4.13). Furthermore, the distance between the positive and negative pairs sample (respectively $\tilde{\boldsymbol{q}}_{\boldsymbol{I}}$ with the queue K_- and $\tilde{\boldsymbol{q}}_{\boldsymbol{T}}$ with the queue K_- are increasing w.r.t L_2 euclidean-distance and converging to 0 w.r.t cosine similarity (Figure 4.14). Once again, we observe that the contrast task induce similar pulling and pushing effect on $\tilde{\boldsymbol{q}}_{\boldsymbol{I}}$ and $\tilde{\boldsymbol{q}}_{\boldsymbol{T}}$.

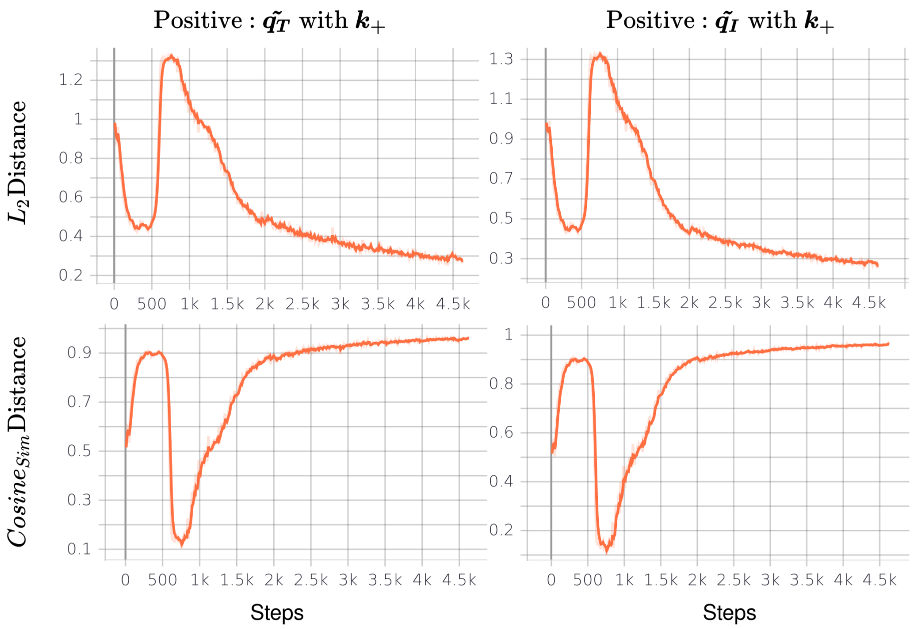


Figure 4.13: Evolution of the similarity score between positive joint-embedding samples $\tilde{\boldsymbol{q}}_{\boldsymbol{I}}$ or $\tilde{\boldsymbol{q}}_{\boldsymbol{T}}$ and K_- when pre-training ViLT with the robust multimodal MoCo frameworks. The similarity score is either compute with the cosine similarity or the L_2 norm.

BarlowTwins

Under the BarlowTwins settings, we observe similar results. The distance between positive sample (respectively $\tilde{\boldsymbol{q}}_{\boldsymbol{I}}$ with $\boldsymbol{k}_+$ and $\tilde{\boldsymbol{q}}_{\boldsymbol{T}}$ with $\boldsymbol{k}_+$ are decreasing w.r.t L_2 euclidean-distance and converging to 1 w.r.t cosine similarity (Figure 4.13). Similarly to the loss analysis (Figure 4.12), the positive distance converges slower than in the Moco case. It reinforce our hypothesis that BarlowTwins under our robust multimodal framework would benefits more from an increase of epochs than MoCo. If we look in detail, we note that the positive distance of $\tilde{\boldsymbol{q}}_{\boldsymbol{I}}$ with the original joint-embedding is slightly bigger than the positive distance of

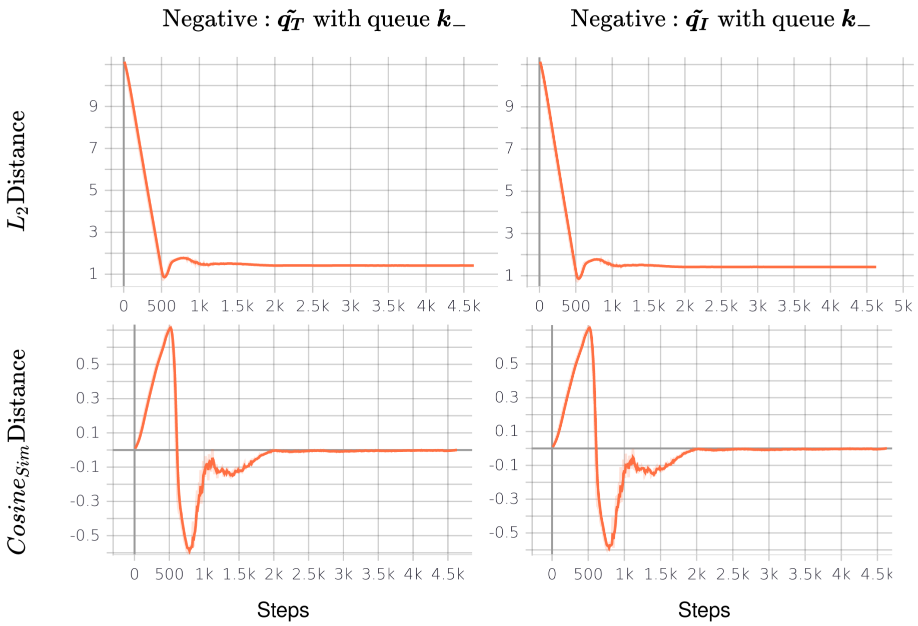
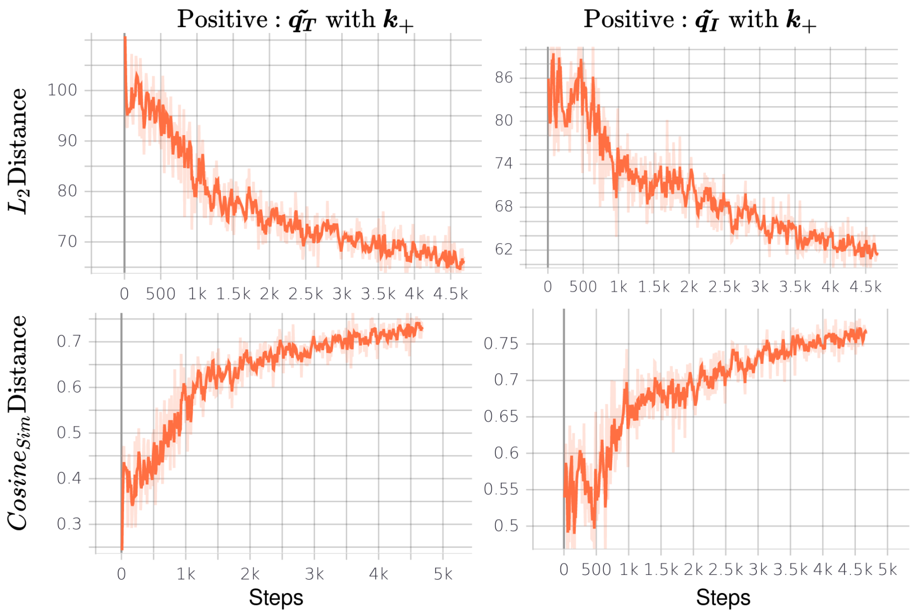


Figure 4.14: Evolution of the similarity score between positive and negative joint-embedding samples $\tilde{q}_I$ or $\tilde{q}_T$ and K_- when pre-training ViLT with the robust multimodal MoCo frameworks. The similarity score is either compute with the cosine similarity or the $L-2$ norm.

$\tilde{q}_T$. It is worth mentioning that we haven't presented the figure of the negative distance for BarlowTwins however the negative distance is behaving as in the MoCo settings.

4.2.3 Adversaries

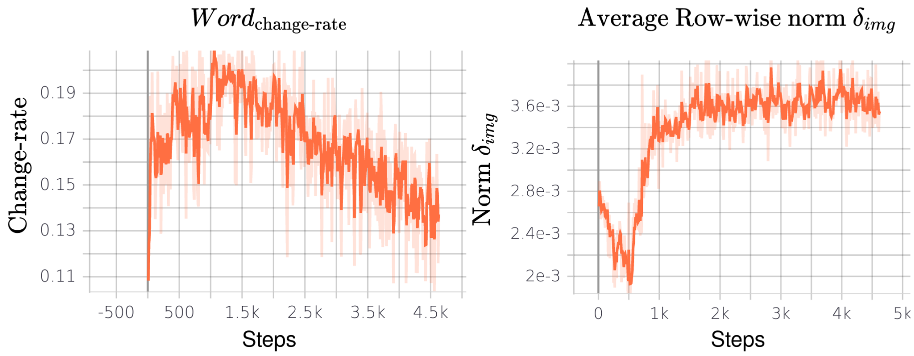
Finally, the the Figure 4.16 and 4.17 illustrate some relevant adversaries metrics. We notice that despite the decrease of loss, the average word change rate per sentence is slightly oscillating around the constant rate of 17.8% for both MoCo and BarlowTwins. This corresponds to the same word change rate discuss in our hyper-parameter search in section 4.1.2. Similarly, the average row-wise norm δ_{img} is oscillating around $3.6e^3$% for MoCo and $2.75e^3$% for BarlowTwins. Suprisingly, the average word change rate of MoCo (Figure 4.16) seems to slightly decreased over time. It would be interesting to observe it's evolution in the next few epochs.



Positive : $\tilde{q_T}$ with $\boldsymbol{k}_+$

Positive : $\tilde{q_I}$ with $\boldsymbol{k}_+$

Figure 4.15: Evolution of the similarity score between positive joint-embedding samples $\tilde{q_I}$ or $\tilde{q_T}$ and K_- when pre-training ViLT with the robust multimodal BarlowTwins frameworks. The similarity score is either compute with the cosine similarity or the $L-2$ norm.



$Word_{\text{change-rate}}$

Average Row-wise norm δ_{img}

Figure 4.16: Evolution of the average row-wise norm δ_{img} and the word change rate when pre-training ViLT with the robust multimodal MoCo frameworks.

4.3 Fine-tuning BarlowTwins and MoCo

We conduct a robustness evaluation on a classification task NLVR2. As discussed 3.3, NLVR2 is an out-of-domain methods. Furthermore, we evaluate performance on image-text retrieval (IRTR) on the in-domain dataset COCO and out-of-domain dataset Flickr30K after fine-tuning.

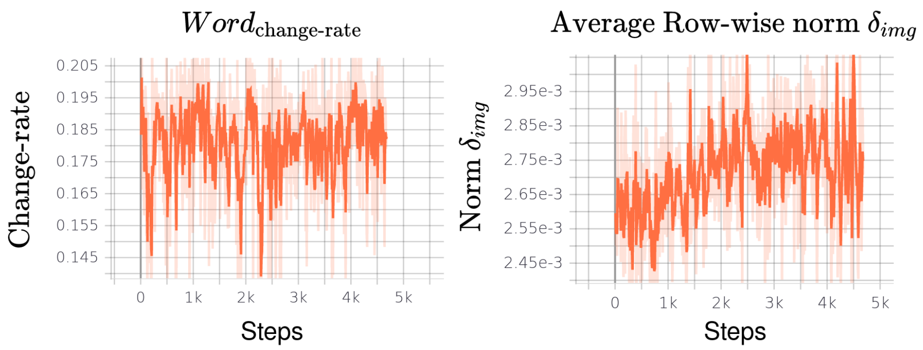


Figure 4.17: Evolution of the average row-wise norm δ_{img} and the word change rate when pre-training ViLT with the robust multimodal BarlowTwins frameworks.

4.3.1 Robustness : NLVR2 Out-of-Domain

The table 4.1 displays the results of the robustness evaluation of NLVR2. ViLT-B/32-NLVR2-paper is the accuracy claim in the original ViLT paper (Kim et al., 2021 [4]), and ViLT-B/32-NLVR2-reimp is our re-implementation. Specifically, we finetune the pre-trained ViLT-B/32 on NLVR2 for 10 epochs. The reimplementation matches the original finetune accuracy. MoCo-NLVR2 and BarlowTwins-NLVR2 are ViLT-B/32 pre-trained on COCO with our RMCL methods with respectively BarlowTwins and MoCo and finetuned on NLVR2 for 10 epoch. We run four types of experience for each model ; (1) Only attacking the pixel space with PGD, (2) Only attacking the words space with the geometric-inspired attack, (3) attacking both image and text space, and (4) making no attack. For each experiment, we report the final test accuracy as well as the success rate of the attack. The attack has been conducted with the exact same adversaries hyper-parameter used in our pre-training stage; for Geometric-inspired attack $M = 5$ and $max_{loop} = 10$ and for PDG $\epsilon = 0.005$ and $\alpha = 0.05$,

	NLVR2							
Model	PGD		GEOM		PGD+GEOM		NONE	
	succ	acc	succ	acc	succ	acc	succ	acc
ViLT-B/32-NLVR2-paper	/	/	/	/	/	/	/	75.7
ViLT-B/32-NLVR2-reimp.	65.9	10.1	45.0	34.0	61.9	9.8	/	**75.6**
MoCo-NLVR2	58.6	**13.9**	39.0	36.7	60.3	10.1	/	72.9
BarlowTwins-NLVR2	63.0	11.1	40.7	**36.9**	64.9	**10.5**	/	75.0

Table 4.1: Robustness evaluation on out-of-domain NLVR2 dataset with different pre-trained models.

We observe that both MoCo and BarlowTwins get better robustness against the PGD attack anf the Geometric-inspired attack. It is interesting to observed that MoCo outperform BarlowTwins on the robustness against PGDs attacks (improvement of 3.8% for moco and 1% for BarlowTwins) while BarlowTwins slightly outperform MoCo on the robustness against Geometric-inspired attacks (improvement of 2.6% for moco and 2.8% for BarlowTwins). We explain this behaviour through the comparison of the MoCo loss (Figure 4.11) versus the BarlowTwins loss (Figure 4.11). Indeed, as discussed in subsection 4.2.1, BarlowTwins seems that it will benefit more from an increase of epochs than MoCo. BarlowTwins's losses delineate a linearly decreasing trend around the end of the first epoch, where MoCo seems to reach a plateau. It suggests that BarlowTwins is delayed compare to MoCo in it's performance. Specifically, if we increase the number of epochs, we expect BarlowTwins to improve its overall performance until it ultimately reached better PGD's robustness than MoCo. In parallel, the asymmetries of robustness performance illustrated in the table correlated with our delayed argument suggest that the PGD's robustness is more challenging to obtain than geometric's robustness. Indeed, even thus BarlowTwins have a slower learning process than MoCo it already outperforms it's geometric's robustness performance. This reasoning is supported by the comparison between the success rate of PDG (Succ = 65.9%) versus Geometric-inspired attacks (Succ = 55.3%) in table 4.1. Apart from these observations, the general accuracy of our model is moderately smaller than the original implementation. This is expected behavior. Indeed, as discussed in section 1.1 adversarial training helps model gain in robustness and generalization but at the cost of slightly lowering the task-specific accuracy.

4.3.2 Robustness : VQAv2 In-Domain

The table 4.2 presents the results of the robustness evaluation of VQAv2. We evaluate the robustness on the validation set instead of the test set as initially done in ViLT (kim et al., 2021[4])paper. The evaluation on the test set require to submit our score on the VQAv2 website, thus we use the validation set. Therefore, we can not compare our re-implementation to the paper claimed accuracy on VQAv2. However, we assume out re-implementation to have identical performance than in the original paper.

	VQAv2			
Model	PGD	GEOM	PGD+GEOM	NONE
	acc	acc	acc	acc
ViLT-B/32-NLVR2-reimp.	80.1	69.1	68.7	**83.5**
MoCo-NLVR2	81.0	73.1	71.8	83.2
BarlowTwins-NLVR2	**81.0**	**73.3**	**72.5**	83.4

Table 4.2: Robustness evaluation on in-domain VQAv2 dataset with different pre-trained models.

We discern that both MoCo and BarlowTwins get better robustness against the PGD attack and the Geometric-inspired attack. The results of robustness on VQA shows that the improvement of robustness on PGD are similar to those from NLVR2. However, we observe that the improvement on the geometric inspired attack is higher on VQAv2 than on NLVR2. As discussed in section 3.4, the implementation of NLVR2 is unlike the pre-training settings ; it required two pairs (Image1, Question1) and (Image2, Question1). It could be then expected that the robustness on NLVR2 is harder than on VQAv2 since the implementation of VQAv2 is similar to the pre-training settings. Additionally, VQAv2 is an in-domain classification task where NLVR2 is out-of-domain. Finally, the results suggest that BarlowTwins's framework get higher performance on VQAv2 than MoCo's frameworks.

4.3.3 Evaluation : IRTR COCO & IRTR Flickr30K

The tables 4.3 and 4.4 lay out the evaluation of the performance on image and text retrieval on COCO (in-domain) and Flickr30K (out-of-domain). As discuss in section 3.4, in image and text retrieval, we sample 15 negative text or images, and we compute the similarity score of each pair. The target of the task is to give the highest similarity score at the original positive pair and lower scores at the negative pairs where one modality has been replaced by a negative sample. We represent the R@1, R@5, and R@10 scores for each model [3]. We use the same pre-trained model as in the previous experience and finetune them either on IRTR-COCO either on IRTR-Flickr30K.

	COCO					
	Image Retrieval			Text Retrieval		
Model	R@1	R@5	R@10	R@1	R@5	R@10
ViLT-B/32-reimp	66.1	92.8	97.2	79.0	95.8	98.4
MoCo-IRTR-COCO	69.4	94.9	98.4	83.8	**98.0**	**99.8**
BarlowTwins-IRTR-COCO	**71.3**	**95.8**	**98.6**	**85.2**	97.2	**99.4**

Table 4.3: Evaluation image and text retrieval on in-domain COCO dataset with different pre-trained models.

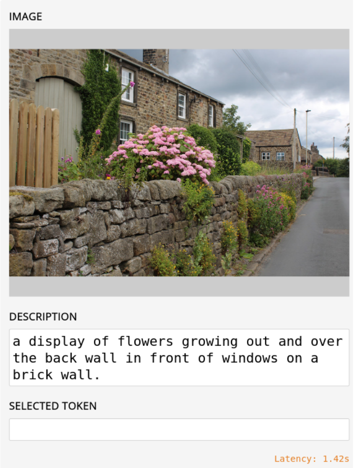
The table 4.3 confirms our intention to reinforce the relation between image and text from the same pair with our robust contrastive task. Indeed, BarloTwins and MoCo shows significant improvement on both retrieval tasks with COCO. However, this improvement are almost nonexistent on Flickr30K (table 4.3). It may shows that pre-training our frameworks only on COCO for a single epochs does not yet build a generalized joint representation enough. Indeed, the matching performance do not get significantly better outside the pre-trained datasets.

[3] R@K corresponds to whether the ground truth is included among top K similarity score.

	Flickr30K					
	Image Retrieval			Text Retrieval		
Model	R@1	R@5	R@10	R@1	R@5	R@10
ViLT-B/32-reimp.	**70.0**	**92.6**	96.4	**88.0**	98.0	**99.8**
MoCo-IRTR-Flickr30K	69.5	91.8	96.7	86.8	98.0	99.4
BarlowTwins-IRTR-Flickr30K	**70.0**	92.5	**96.8**	87.8	**98.2**	99.6

Table 4.4: Evaluation image and text retrieval on out-of-domain Flickr30K dataset with different pre-trained models.

4.4 Out-of-the-box MLM visualization

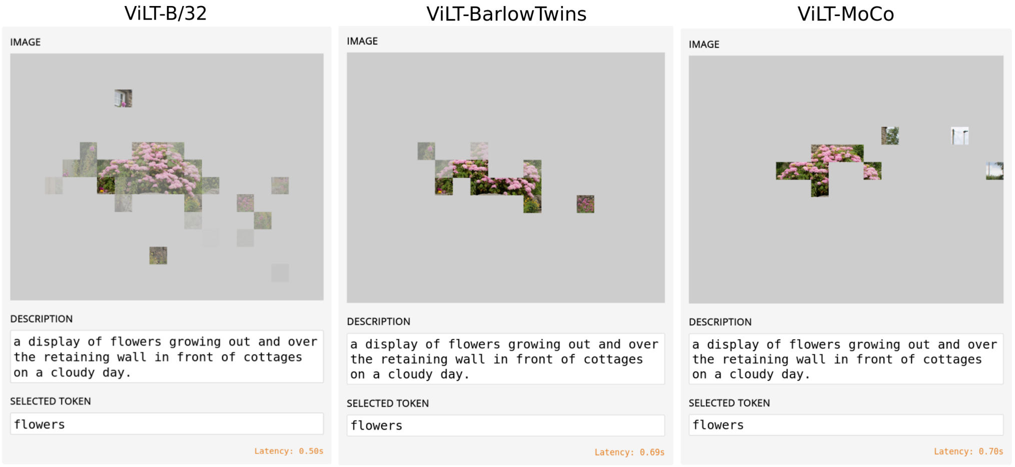


a display of flowers growing out and over the back wall in front of windows on a brick wall.

Figure 4.18: Out-of-domain Image text pair.

The figures 4.18 and 4.19 illustrate an example of a cross-modal alignment. The transportation plan of the IPOT (see section 2.3.2) displays a heatmap for a selected text token. Each square tile describes a patch, and its opacity designates how much quantity is carried from the highlighted word token.The figure 4.18 represent an out-of-domain image-text pair and the figure 4.19 illustrate the results for ViLT-MoCo and ViLT-BarlowTwins and ViLT-B/32.

The figure 4.19 suggests that our robust multimodal contrastive task reinforce the image and text matching. Although, ViLT-B/32 has a great ability to match the different patches to a selected token, both ViLT-MoCo and ViLT-BarlowTwins reinforce its image-text matching intensity. Indeed, we can observe that the patches transported from the selected token are much more opaque that in the standard pre-trained ViLT model.



ViLT-B/32

IMAGE

DESCRIPTION

a display of flowers growing out and over
the retaining wall in front of cottages
on a cloudy day.

SELECTED TOKEN

flowers

Latency: 0.50s

ViLT-BarlowTwins

IMAGE

DESCRIPTION

a display of flowers growing out and over
the retaining wall in front of cottages
on a cloudy day.

SELECTED TOKEN

flowers

Latency: 0.69s

ViLT-MoCo

IMAGE

DESCRIPTION

a display of flowers growing out and over
the retaining wall in front of cottages
on a cloudy day.

SELECTED TOKEN

flowers

Latency: 0.70s

Figure 4.19: Visualizations of transportation plan of word patch alignment with three different pretained ViLT model.

Chapter 5

Discussion & Conclussion

In this thesis, we present RMCL, a robust multimodal contrastive learning task. We verify the efficacy of our proposed method by leveraging two well-known contrastive frameworks MoCo (Chen et al., 2020 [3]) and BarlowTwins (Zbontar et al., 2021 [2]) in a robust multimodal setting. We used a pre-trained V&L model ViLT as an encoder and compare it's robustness and accuracy performance with and without our RMCL methods. Before applying our pre-trained methods on ViLT, we conduct a hyperparameter selection by attacking the NLVR2 evaluation process of ViLT. Once our hyper-parameter is selected, we start pre-training ViLT with RMCL. Due to time constraints, we only used the COCO dataset with one epoch for pre-training all of our settings. However, it already gives a decent overview of our task behaviors. Specifically, during the training phase, our RMCL task successfully learns to pull together the joint-embedding of an image-text pair with its attacked counterpart while pushing apart the dissimilar joint-embedding pairs. Next, we have evaluate the robustness and accuracy performance of our RMCL tasks on ViLT with an out-of-domain (NLVR2) and in-domain (VQAv2) classification tasks. The results suggest that our methods lead the ViLT model to get better robustness against text and image attacks while having a slightly lower accuracy than the standardly-trained ViLT. Although the improvement is minor, it suggests that our task behave correctly. Indeed, we can observe in tables 3.2 and 3.1 that COCO is only slightly bigger than the downstream datasets. Futhermore, improvement of robustness in both in-domain and out-of-domain dataset confirms that our robust contrastive optimization drives the model to get a more generalized and robust joint representation.

Next, we evaluate the performance of the RMCL pre-trained ViLT on image and text retrieval. We observed a significant improvement with our methods over the standard ViLT on an in-domain dataset. However, the retrieval performance on the out-of-domain dataset is almost nonexistent, although competitive. It points out that our methodology leads to more generalized joint representation so that the image and text relation gets enhanced. However, the generalization doesn't go yet outside the pre-training datasets. We advocate that pre-training on only MoCo for a single epoch is not yet enough for a broad generalized joint representation. Finally, our experience demonstrates that BarlowTwins achieves usually better overall performance over MoCo under our RMCL settings. Furthermore, the losses analysis (4.12) suggest that BarlowTwins should benefits more than MoCo from an increased of epochs.

The above considerations and experiences compose our conjecture that the joint representation of an image text pair benefits from the robust optimization of a multimodal contrastive learning

In the future of this project, we will extend the size of the dataset and the number of epochs for pre-training as it was initially planned. To be specific we will pre-trained our model for 10 epochs on Microsoft COCO (Lin et al., 2014 [60]), Visual Genome (VG) (Krishna et al., 2017 [61]), SBU Captions (SBU) (Ordonez et al., 2011 [62]), and Google Conceptual Captions (GCC) (Sharma et al., [63]). We will then conduct the same sets of experience on the downstream tasks. We will as well conduct a full pretraining on our MCL baseline and compare their generalization and robustness performances.

In parallel, we will examined the uses of stronger and synonymous more perceptible adversaries. Indeed, as discuss in section 2.5 the contrastive tasks in general benefits more from strong augmentation provided that it keeps the overall semantic of the sample. It may also be possible that it benefits the robustness performances.

Apart from theses improvements, we advocate that doing the image-text matching (ITM) pre-training methods afterward, having done our RMCL tasks, will be a great benefit. Indeed, when our methods is further fine tuned on retrieval tasks we obtain better results on both image and text retrieval task with in-domain dataset. Our methods not only make the the joint-representation invariant to perturbation it also reinforce the image and text relations. This though is supported by our experience on transportation plan of word patch alignment. Indeed, the image-text relation within a same par is visually strengthened.

Bibliography

[1] A. Ilyas, S. Santurkar, D. Tsipras, L. Engstrom, B. Tran, and A. Madry, "Adversarial examples are not bugs, they are features," *arXiv preprint arXiv:1905.02175*, 2019.

[2] J. Zbontar, L. Jing, I. Misra, Y. LeCun, and S. Deny, "Barlow twins: Self-supervised learning via redundancy reduction," *arXiv preprint arXiv:2103.03230*, 2021.

[3] X. Chen, H. Fan, R. Girshick, and K. He, "Improved baselines with momentum contrastive learning," *arXiv preprint arXiv:2003.04297*, 2020.

[4] W. Kim, B. Son, and I. Kim, "Vilt: Vision-and-language transformer without convolution or region supervision," *arXiv preprint arXiv:2102.03334*, 2021.

[5] Richardson and J. Vectors, "aphorisms ten-second essays," in *Ausable Press*, Jun. 2001.

[6] D. H. Wolpert and W. G. Macready, "No free lunch theorems for optimization," *IEEE transactions on evolutionary computation*, vol. 1, no. 1, pp. 67–82, 1997.

[7] R. Geirhos, J.-H. Jacobsen, C. Michaelis, R. Zemel, W. Brendel, M. Bethge, and F. A. Wichmann, "Shortcut learning in deep neural networks," *Nature Machine Intelligence*, vol. 2, no. 11, pp. 665–673, 2020.

[8] S. Gidaris, P. Singh, and N. Komodakis, "Unsupervised representation learning by predicting image rotations," *arXiv preprint arXiv:1803.07728*, 2018.

[9] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, "Bert: Pre-training of deep bidirectional transformers for language understanding," *arXiv preprint arXiv:1810.04805*, 2018.

[10] Y. Liu, M. Ott, N. Goyal, J. Du, M. Joshi, D. Chen, O. Levy, M. Lewis, L. Zettlemoyer, and V. Stoyanov, "Roberta: A robustly optimized bert pretraining approach," *arXiv preprint arXiv:1907.11692*, 2019.

[11] A. Conneau, K. Khandelwal, N. Goyal, V. Chaudhary, G. Wenzek, F. Guzmán, E. Grave, M. Ott, L. Zettlemoyer, and V. Stoyanov, "Unsupervised cross-lingual representation learning at scale," *arXiv preprint arXiv:1911.02116*, 2019.

[12] T. Chen, S. Kornblith, M. Norouzi, and G. Hinton, “A simple framework for contrastive learning of visual representations,” in *International conference on machine learning.* PMLR, 2020, pp. 1597–1607.

[13] J.-B. Grill, F. Strub, F. Altché, C. Tallec, P. H. Richemond, E. Buchatskaya, C. Doersch, B. A. Pires, Z. D. Guo, M. G. Azar *et al.*, “Bootstrap your own latent: A new approach to self-supervised learning,” *arXiv preprint arXiv:2006.07733*, 2020.

[14] Y. Tian, C. Sun, B. Poole, D. Krishnan, C. Schmid, and P. Isola, “What makes for good views for contrastive learning?” *arXiv preprint arXiv:2005.10243*, 2020.

[15] A. Radford, J. W. Kim, C. Hallacy, A. Ramesh, G. Goh, S. Agarwal, G. Sastry, A. Askell, P. Mishkin, J. Clark *et al.*, “Learning transferable visual models from natural language supervision,” *arXiv preprint arXiv:2103.00020*, 2021.

[16] J. Gilmer, L. Metz, F. Faghri, S. S. Schoenholz, M. Raghu, M. Wattenberg, and I. Goodfellow, “Adversarial spheres,” *arXiv preprint arXiv:1801.02774*, 2018.

[17] T. Tanay and L. Griffin, “A boundary tilting persepective on the phenomenon of adversarial examples,” *arXiv preprint arXiv:1608.07690*, 2016.

[18] L. Schmidt, S. Santurkar, D. Tsipras, K. Talwar, and A. Mądry, “Adversarially robust generalization requires more data,” *arXiv preprint arXiv:1804.11285*, 2018.

[19] H. Salman, A. Ilyas, L. Engstrom, A. Kapoor, and A. Madry, “Do adversarially robust imagenet models transfer better?” *arXiv preprint arXiv:2007.08489*, 2020.

[20] T. Chen, S. Liu, S. Chang, Y. Cheng, L. Amini, and Z. Wang, “Adversarial robustness: From self-supervised pre-training to fine-tuning,” in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2020, pp. 699–708.

[21] M. Kim, J. Tack, and S. J. Hwang, “Adversarial self-supervised contrastive learning,” *arXiv preprint arXiv:2006.07589*, 2020.

[22] Z. Meng, Y. Dong, M. Sachan, and R. Wattenhofer, “Self-supervised contrastive learning with adversarial perturbations for robust pretrained language models,” *arXiv preprint arXiv:2107.07610*, 2021.

[23] A. Madry, A. Makelov, L. Schmidt, D. Tsipras, and A. Vladu, “Towards deep learning models resistant to adversarial attacks,” *arXiv preprint arXiv:1706.06083*, 2017.

[24] Z. Meng and R. Wattenhofer, "A geometry-inspired attack for generating natural language adversarial examples," *arXiv preprint arXiv:2010.01345*, 2020.

[25] A. Dosovitskiy, L. Beyer, A. Kolesnikov, D. Weissenborn, X. Zhai, T. Unterthiner, M. Dehghani, M. Minderer, G. Heigold, S. Gelly *et al.*, "An image is worth 16x16 words: Transformers for image recognition at scale," *arXiv preprint arXiv:2010.11929*, 2020.

[26] A. Suhr, S. Zhou, A. Zhang, I. Zhang, H. Bai, and Y. Artzi, "A corpus for reasoning about natural language grounded in photographs," *arXiv preprint arXiv:1811.00491*, 2018.

[27] Y. Goyal, T. Khot, D. Summers-Stay, D. Batra, and D. Parikh, "Making the v in vqa matter: Elevating the role of image understanding in visual question answering," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2017, pp. 6904–6913.

[28] A. Karpathy and L. Fei-Fei, "Deep visual-semantic alignments for generating image descriptions," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2015, pp. 3128–3137.

[29] A. Bandura, "Social learning theory," *New York: General Learning Press, 197130*, 1991.

[30] A. v. d. Oord, N. Kalchbrenner, O. Vinyals, L. Espeholt, A. Graves, and K. Kavukcuoglu, "Conditional image generation with pixelcnn decoders," *arXiv preprint arXiv:1606.05328*, 2016.

[31] Z. Lan, M. Chen, S. Goodman, K. Gimpel, P. Sharma, and R. Soricut, "Albert: A lite bert for self-supervised learning of language representations," *arXiv preprint arXiv:1909.11942*, 2019.

[32] J. Bromley, J. W. Bentz, L. Bottou, I. Guyon, Y. LeCun, C. Moore, E. Säckinger, and R. Shah, "Signature verification using a "siamese" time delay neural network," *International Journal of Pattern Recognition and Artificial Intelligence*, vol. 7, no. 04, pp. 669–688, 1993.

[33] A. v. d. Oord, Y. Li, and O. Vinyals, "Representation learning with contrastive predictive coding," *arXiv preprint arXiv:1807.03748*, 2018.

[34] Z. Wu, Y. Xiong, S. Yu, and D. Lin, "Unsupervised feature learning via non-parametric instance-level discrimination," *arXiv preprint arXiv:1805.01978*, 2018.

[35] F. Wang, X. Xiang, J. Cheng, and A. L. Yuille, "Normface: L2 hypersphere embedding for face verification," in *Proceedings of the 25th ACM international conference on Multimedia*, 2017, pp. 1041–1049.

[36] T. Wang and P. Isola, "Understanding contrastive representation learning through alignment and uniformity on the hypersphere," in *International Conference on Machine Learning.* PMLR, 2020, pp. 9929–9939.

[37] X. Chen and K. He, "Exploring simple siamese representation learning," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2021, pp. 15 750–15 758.

[38] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, and I. Polosukhin, "Attention is all you need," in *Advances in neural information processing systems*, 2017, pp. 5998–6008.

[39] H. Tan and M. Bansal, "Lxmert: Learning cross-modality encoder representations from transformers," *arXiv preprint arXiv:1908.07490*, 2019.

[40] J. Lu, D. Batra, D. Parikh, and S. Lee, "Vilbert: Pretraining task-agnostic visiolinguistic representations for vision-and-language tasks," *arXiv preprint arXiv:1908.02265*, 2019.

[41] Y.-C. Chen, L. Li, L. Yu, A. El Kholy, F. Ahmed, Z. Gan, Y. Cheng, and J. Liu, "Uniter: Learning universal image-text representations," 2019.

[42] L. H. Li, M. Yatskar, D. Yin, C.-J. Hsieh, and K.-W. Chang, "Visualbert: A simple and performant baseline for vision and language," *arXiv preprint arXiv:1908.03557*, 2019.

[43] N. O. Emanuele Bugliarello, Ryan Cotterell and D. Elliott, "ultimodal pretraining unmasked: A meta-analysis and a unified framework of vision-and-language berts," *arXiv:2011.15124v2*, 2021.

[44] S. Ren, K. He, R. Girshick, and J. Sun, "Faster r-cnn: Towards real-time object detection with region proposal networks," *Advances in neural information processing systems*, vol. 28, pp. 91–99, 2015.

[45] J. Cho, J. Lu, D. Schwenk, H. Hajishirzi, and A. Kembhavi, "X-lxmert: Paint, caption and answer questions with multi-modal transformers," *arXiv preprint arXiv:2009.11278*, 2020.

[46] Z. Huang, Z. Zeng, B. Liu, D. Fu, and J. Fu, "Pixel-bert: Aligning image pixels with text by deep multi-modal transformers," *arXiv preprint arXiv:2004.00849*, 2020.

[47] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 770–778.

[48] Y. Xie, X. Wang, R. Wang, and H. Zha, "A fast proximal point method for computing exact wasserstein distance," in *Uncertainty in Artificial Intelligence.* PMLR, 2020, pp. 433–453.

[49] A. Dosovitskiy, J. T. Springenberg, M. Riedmiller, and T. Brox, "Discriminative unsupervised feature learning with convolutional neural networks," *Advances in neural information processing systems*, vol. 27, pp. 766–774, 2014.

[50] Z. Jiang, T. Chen, T. Chen, and Z. Wang, "Robust pre-training by adversarial contrastive learning." in *NeurIPS*, 2020.

[51] H. Fang, S. Wang, M. Zhou, J. Ding, and P. Xie, "Cert: Contrastive self-supervised learning for language understanding," *arXiv preprint arXiv:2005.12766*, 2020.

[52] R. Sennrich, B. Haddow, and A. Birch, "Improving neural machine translation models with monolingual data," *arXiv preprint arXiv:1511.06709*, 2015.

[53] J. Wei and K. Zou, "Eda: Easy data augmentation techniques for boosting performance on text classification tasks," *arXiv preprint arXiv:1901.11196*, 2019.

[54] J. Zhang, Y. Zhao, M. Saleh, and P. Liu, "Pegasus: Pre-training with extracted gap-sentences for abstractive summarization," in *International Conference on Machine Learning.* PMLR, 2020, pp. 11 328–11 339.

[55] N. Reimers and I. Gurevych, "Sentence-bert: Sentence embeddings using siamese bert-networks," *arXiv preprint arXiv:1908.10084*, 2019.

[56] Z. Gan, Y.-C. Chen, L. Li, C. Zhu, Y. Cheng, and J. Liu, "Large-scale adversarial training for vision-and-language representation learning," *arXiv preprint arXiv:2006.06195*, 2020.

[57] M. Alzantot, Y. Sharma, A. Elgohary, B.-J. Ho, M. Srivastava, and K.-W. Chang, "Generating natural language adversarial examples," *arXiv preprint arXiv:1804.07998*, 2018.

[58] S. Ren, Y. Deng, K. He, and W. Che, "Generating natural language adversarial examples through probability weighted word saliency," in *Proceedings of the 57th annual meeting of the association for computational linguistics*, 2019, pp. 1085–1097.

[59] D. Jin, Z. Jin, J. T. Zhou, and P. Szolovits, "Is bert really robust? a strong baseline for natural language attack on text classification and entailment," in *Proceedings of the AAAI conference on artificial intelligence*, vol. 34, no. 05, 2020, pp. 8018–8025.

[60] T.-Y. Lin, M. Maire, S. Belongie, J. Hays, P. Perona, D. Ramanan, P. Dollár, and C. L. Zitnick, "Microsoft coco: Common objects in context," in *European conference on computer vision.* Springer, 2014, pp. 740–755.

[61] R. Krishna, Y. Zhu, O. Groth, J. Johnson, K. Hata, J. Kravitz, S. Chen, Y. Kalantidis, L.-J. Li, D. A. Shamma *et al.*, "Visual genome: Connecting language and vision using crowdsourced dense image annotations," *arXiv preprint arXiv:1602.07332*, 2016.

[62] V. Ordonez, G. Kulkarni, and T. Berg, "Im2text: Describing images using 1 million captioned photographs," *Advances in neural information processing systems*, vol. 24, pp. 1143–1151, 2011.

[63] P. Sharma, N. Ding, S. Goodman, and R. Soricut, "Conceptual captions: A cleaned, hypernymed, image alt-text dataset for automatic image captioning," in *Proceedings of ACL*, 2018.

[64] Z. Yu, J. Yu, Y. Cui, D. Tao, and Q. Tian, "Deep modular co-attention networks for visual question answering," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2019, pp. 6281–6290.

[65] X. Li, X. Yin, C. Li, P. Zhang, X. Hu, L. Zhang, L. Wang, H. Hu, L. Dong, F. Wei *et al.*, "Oscar: Object-semantics aligned pre-training for vision-language tasks," in *European Conference on Computer Vision.* Springer, 2020, pp. 121–137.

[66] P. Zhang, X. Li, X. Hu, J. Yang, L. Zhang, L. Wang, Y. Choi, and J. Gao, "Vinvl: Revisiting visual representations in vision-language models," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2021, pp. 5579–5588.

[67] P. Micikevicius, S. Narang, J. Alben, G. Diamos, E. Elsen, D. Garcia, B. Ginsburg, M. Houston, O. Kuchaiev, G. Venkatesh *et al.*, "Mixed precision training," *arXiv preprint arXiv:1710.03740*, 2017.

[68] S. Li, Y. Zhao, R. Varma, O. Salpekar, P. Noordhuis, T. Li, A. Paszke, J. Smith, B. Vaughan, P. Damania *et al.*, "Pytorch distributed: Experiences on accelerating data parallel training," *arXiv preprint arXiv:2006.15704*, 2020.

[69] M. Ott, S. Edunov, D. Grangier, and M. Auli, "Scaling neural machine translation," *arXiv preprint arXiv:1806.00187*, 2018.

Appendix A
Algorithm

Algorithm 1: Robust Multi-Modal Contrastive Learning (RMCL) under MoCo settings

```
Inputs: Set of Image-Text pairs $\{(I,T)\}$, batch Size $N$, temperature $\tau$,
        $\epsilon$-Ball, number of synonym $M$, max step PGD I, max step Geom J
        momentum $m$, queue of negative joint embeddings $K_-$,
        structure of query and key multimodal encoder $f_\theta$, $f_\xi$
        structure of query and key projector $g_\theta$, $g_\xi$
Forward pass
for all $x \in$ Minibatch $B = \{(I_i, T_i)\}_{i=1}^{N}$ do
    # Generate Instance Wise Attack for Image :
    for $t$= 1...I do
        $\tilde{I_i}^{t+1} = \Pi_{B(I_i,\epsilon)} \left(\tilde{I_i}^{t} + \alpha \,\mathrm{sign}\left(\nabla_{\tilde{q}_i} \mathcal{L}_{\text{RMInfoNCE}}\left(\tilde{q}_i^t, k_i^+, k_i^-\right)\right)\right)$
    end
    # Generate Instance Wise Attack for Text :
    for $t$= 1...J do
        $\tilde{T}_i^t$ = Geometrical-based-attack($T_i, f_\theta, g_\theta, M$)
    end
    # Joint representation and joint embedding of $\left(I_i, \tilde{T}_i\right)$:
    $C\tilde{L}S_i^q$ = Pooler $\left(\text{Encoder}\left((I_i, \tilde{T}_i)\right)\right)$ = Pooler $\left(f_\theta\left((I_i, \tilde{T}_i)\right)\right)$
    $\tilde{q}_i{}^T$ = MLP $(C\tilde{L}S_i^q) = g_\theta(C\tilde{L}S_i^q)$
    # Joint representation and joint embedding of $\left(\tilde{I}_i, T_i\right)$:
    $C\tilde{L}S_i^q$ = Pooler $\left(\text{Encoder}\left((\tilde{I}_i, T_i)\right)\right)$ = Pooler $\left(f_\theta\left((\tilde{I}_i, T_i)\right)\right)$
    $\tilde{q}_i{}^I$ = MLP $(C\tilde{L}S_i^q) = g_\theta(C\tilde{L}S_i^q)$
end
Calculate $L_{RMinfoNCE} = \sum_{\tilde{q} \in \{\tilde{q_I}, \tilde{q_T}\}} -\log \frac{\exp\left(\tilde{q} * k^+ / \tau\right)}{\exp\left(\tilde{q} * k^+ / \tau\right) + \sum_{K^-} \exp\left(\tilde{q} * k^- / \tau\right)}$
Backward pass
# SGD update: query network
$\theta \leftarrow \theta - \eta \cdot \nabla_\theta L_{RMinfoNCE}\left(\tilde{q_I}^{(i:i+N)}; \tilde{q_T}^{(i:i+N)}; k_+^{(i:i+N)}; K_-\right)$
# Momentum update: key network
$\xi \leftarrow m\xi + (1-m)\theta$
# update dictionary
enqueue(queue, k) # enqueue the current minibatch
dequeue(queue) # dequeue the earliest minibatch
```

ETH
Eidgenössische Technische Hochschule Zürich
Swiss Federal Institute of Technology Zurich

Declaration of originality

The signed declaration of originality is a component of every semester paper, Bachelor's thesis, Master's thesis and any other degree paper undertaken during the course of studies, including the respective electronic versions.

Lecturers may also require a declaration of originality for other written papers compiled for their courses.

I hereby confirm that I am the sole author of the written work here enclosed and that I have compiled it in my own words. Parts excepted are corrections of form and content by the supervisor.

Title of work (in block letters):

RMCL : A ROBUST MULTIMODAL CONTRASTIVE LEARNING FRAMEWORK

Authored by (in block letters):

For papers written by groups the names of all authors are required.

Name(s):	**First name(s):**
FURRER	STANISLAS

With my signature I confirm that

- I have committed none of the forms of plagiarism described in the 'Citation etiquette' information sheet.
- I have documented all methods, data and processes truthfully.
- I have not manipulated any data.
- I have mentioned all persons who were significant facilitators of the work.

I am aware that the work may be screened electronically for plagiarism.

Place, date

Zurich, 02.08.2021

Signature(s)

For papers written by groups the names of all authors are required. Their signatures collectively guarantee the entire content of the written paper.