HOCHSCHULE
HANNOVER
UNIVERSITY OF
APPLIED SCIENCES
AND ARTS
–
Fakultät IV
Wirtschaft und
Informatik

Titel der Arbeit

Tim Cares

Master's thesis in Applied Computer Science

25. September 2024

Autor Tim Cares
Matrikelnummer
Email Adresse

Erstprüfer Prof. Dr. Vorname Name
Abteilung Informatik, Fakultät IV
Hochschule Hannover
Email Adresse

Zweitprüfer Prof. Dr. Vorname Name
Abteilung Informatik, Fakultät IV
Hochschule Hannover
Email Adresse

I hereby declare that I have written and submitted this thesis independently, without any external help or use of sources and aids other than those specifically mentioned by me. I also declare that I have not taken any content from the works used without proper citation and acknowledgement.

Hannover, 25. September 2024 Tim Cares

Acknowledgements

Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod tempor incididunt ut labore et dolore magnam aliquam quaerat voluptatem. Ut enim aeque doleamus animo, cum corpore dolemus, fieri tamen permagna accessio potest, si aliquod aeternum et infinitum impendere malum nobis opinemur. Quod idem licet transferre in voluptatem, ut postea variari voluptas distinguique possit, augeri amplificarique non possit. At etiam Athenis, ut e patre audiebam facete et urbane Stoicos irridente, statua est in quo a nobis philosophia defensa et collaudata est, cum id, quod maxime placeat, facere possimus, omnis voluptas assumenda est, omnis dolor repellendus. Temporibus autem quibusdam et aut officiis debitis aut rerum necessitatibus saepe eveniet, ut et voluptates repudiandae sint et molestiae non recusandae. Itaque earum rerum defuturum, quas natura non depravata desiderat. Et quem ad me accedis, saluto: 'chaere,' inquam, 'Tite!' lictores, turma omnis chorusque: 'chaere, Tite!' hinc hostis mi Albucius, hinc inimicus. Sed iure Mucius. Ego autem mirari satis non queo unde hoc sit tam insolens domesticarum rerum fastidium. Non est omnino hic docendi locus; sed ita prorsus existimo, neque eum Torquatum, qui hoc primus cognomen invenerit, aut torquem illum hosti detraxisse, ut aliquam ex eo est consecutus? – Laudem et caritatem, quae sunt vitae.

Abstract

Note:

1. **paragraph:** What is the motivation of your thesis? Why is it interesting from a scientific point of view? Which main problem do you like to solve?
2. **paragraph:** What is the purpose of the document? What is the main content, the main contribution?
3. **paragraph:** What is your methodology? How do you proceed?

Zusammenfassung

Note: Insert the German translation of the English abstract here.

Contents

1 Example with Lorem Ipsum

Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod tempor incididunt ut labore et dolore magnam aliquam quaerat voluptatem. Ut enim aeque doleamus animo, cum corpore dolemus, fieri tamen permagna accessio potest, si aliquod aeternum et infinitum impendere malum nobis opinemur. Quod idem licet transferre in voluptatem, ut postea variari voluptas distinguique possit, augeri amplificarique non possit. At etiam Athenis, ut e patre audiebam facete et urbane Stoicos irridente, statua est in quo a nobis philosophia defensa et collaudata est, cum id, quod maxime placeat, facere possimus, omnis voluptas assumenda est, omnis dolor repellendus. Temporibus autem quibusdam et aut officiis debitis aut rerum necessitatibus saepe eveniet, ut et voluptates repudiandae sint et molestiae non recusandae. Itaque earum rerum defuturum, quas natura non depravata desiderat. Et quem ad me accedis, saluto: 'chaere,' inquam, 'Tite!' lictores, turma omnis chorusque: 'chaere, Tite!' hinc hostis mi Albucius, hinc inimicus. Sed iure Mucius. Ego autem mirari satis non queo unde hoc sit tam insolens domesticarum rerum fastidium. Non est omnino hic docendi locus; sed ita prorsus existimo, neque eum Torquatum, qui hoc primus cognomen invenerit, aut torquem illum hosti detraxisse, ut aliquam ex eo est consecutus? – Laudem et caritatem, quae sunt vitae sine metu degendae praesidia firmissima. – Filium morte multavit. – Si sine causa, nollem me ab eo delectari, quod ista Platonis, Aristoteli, Theophrasti orationis ornamenta neglexerit. Nam illud quidem physici, credere aliquid esse minimum, quod profecto numquam putavisset, si a Polyaeno, familiari suo, geometrica discere maluisset quam illum etiam ipsum dedocere. Sol Democrito magnus videtur, quippe homini erudito in geometriaque perfecto, huic pedalis fortasse; tantum enim esse omnino in nostris poetis aut inertissimae segnitiae est aut fastidii delicatissimi. Mihi quidem videtur, inermis ac nudus est. Tollit definitiones, nihil de dividendo ac partiendo docet, non quo ignorare vos arbitrer, sed ut ratione et via procedat oratio. Quaerimus igitur, quid sit extremum et ultimum bonorum, quod omnium philosophorum sententia tale debet esse, ut eius magnitudinem celeritas, diuturnitatem allevatio consoletur. Ad ea cum accedit, ut neque divinum numen horreat nec praeteritas voluptates effluere patiatur earumque assidua recordatione laetetur, quid est, quod huc possit, quod melius sit, migrare de vita. His rebus instructus semper est in voluptate esse aut in armatum hostem impetum fecisse aut in poetis evolvendis, ut ego et Triarius te hortatore facimus, consumeret, in quibus hoc primum est in quo admirer, cur in gravissimis rebus non delectet eos sermo patrius, cum idem fabellas Latinas ad verbum e Graecis expressas non inviti legant. Quis enim tam inimicus paene nomini Romano est, qui Ennii Medeam aut An-

tiopam Pacuvii spernat aut reiciat, quod se isdem Euripidis fabulis delectari dicat, Latinas litteras oderit? Synephebos ego, inquit, potius Caecilii aut Andriam Terentii quam utramque Menandri legam?

2 Introduction

Note: Introduce the topic of your thesis, e.g. with a little historical overview.

List of Figures

List of Tables

2.1 Methodology

2.1.1 Tools

Software:

- for all implementations we use pytorch lightning
- provides high-level functionalities on top of pytorch
  - like checkpointing, logging, distributed training, etc.
- we do not need to implement them in pytorch manually (to some extend)

  - pytorch already provides a high-level API but it is more prone to errors
- we save time and can focus on the actual implementation
- errors in vanilla pytorch are likely and hard to debug
- research will enevitably involve a lot of trial and error (experimentation)
- to keep track of all experiments, we use the experiment tracking tool Weights & Biases

Hardware:
- it is not possible to train the models, used in this work, on the CPU

-> GPUs are a requirement
- should be relatively new -> should be able to handle models upwards of 50 million parameters, but should not be too expensive
- we are severely limited by financial constraints, as there is not external funding for this work
- GPUs rented in the cloud
- we do not use popular cloud services like AWS or GCP -> too expensive
- instead, we use the smaller provider runpod.io
- has a high variety of consumer-grade, and enterprise-grade GPUs, much more affordable
- we opt for the (consumer-grade) NVIDIA RTX 4090
  - has one of the highest speeds (TODO: cite?) but lacks high VRAM (only 24GB)
    - is a problem we will address later
  - at the time of this research (June 2024), comes in at around 0.75 USD per hour
  - higher VRAM GPUs, like the A100, are available for 1.89 USD per hour
    - too expensive in the long run

2.1.2 Experimental Approach

- we will start as simple as possible
- always build on the results and knowledge of the previous steps
- to first validate if Knowledge-Distillation, the approach we will use throughout this work, even works for us, we will first test KD of unimodal models (e.g. distilling a ResNet-50 from a ResNet-101 on ImageNet), an area which has already been researched extensively
- from this, we will advance to the actual goal of this work: Multimodal Knowledge-Distillation
- as this is increasingly more difficult than distilling a unimodal model from another unimodal model of the same architecture, we will start with a supervised teacher
  - means, the teacher model has been trained on labeled data, and provides us with logits, and therefore a probabilty distribution, to regress
    - is basically a reproduction of SHRe [AVT17]
    - has been proven to work with this paper as a proof-of-concept
- if this approach works likewise for us, we will advance to a self-supervised teacher

- recall that goal was build a model/procedure for multimodal KD completly unreliant on labeled data
  - also means teacher, or any pretrained module that might be used, can't be trained on labeled data
  - goal of this work is to check if this is possible
  - as mentioned before, VLMo for example use a BEiT module pretrained on labeled data as part of their model
    - this is not end-to-end self-supervised

2.1.3 Data Collection and Preprocessing

General:

- data we need to collect has to be both unimodal and multimodal
  - multimodal obvious -> needed to align modalities, as described in e.g. section about "(See, Hear, and Read:) Deep Aligned Representations"
  - multimodal means in this case dataset of image-text pairs
  - unimodal -> needed for first tests and poc of distillation process
  - unimodal data also needed for evaluation and comparison of unimodal models from research papers, like Data2Vec, as well as comparison between unimodal and multimodal distilled models (models of this thesis)
  - unimodal data also needed for stage-wise knowledge distillation in section about "Mixing Positional Encodings" (will be elaborated on in the respective section)

Data Selection and Collection:

- starting with unimodal:
- collecting unimodal data is not a problem, many highly curated and large datasets available
- because we are using Knowledge-Distillation based on self-supervised distillation, so we do not need labels for the distillation process, we can use any image dataset
- for image data, we select imagenet
  - build for image classification and object detection (labled, but, again, not necessary)
  - each image corresponds to one of 1k classes
  - very popular, high quality, high variety -> 1000 classes, by standards of SOTA models it is a medium sized dataset (ca. 1.2M train images)
    - why medium sized? -> papers used in this thesis have been trained on much larger data -> VLMo around 14 million image(-text) examples,

    BEiT on more than 35M, and FLAVA (only mentioned a couple of times) even 70M

  - models also much larger than models build here, so we do not need as much, nor is it feasible for us to train on that much data
- Data2Vec Image model exclusively trained on imagenet
- Data2Vec, BEiT, VLMo, and FLAVA all use imagenet for evaluation -> we should use it as well
- we use the full dataset of the 2012 version, with 1.2M images for training and 50k for validation
- during pretraining, i.e. the Knowledge-Distillation, we apply the same data augmentation as in Data2Vec2
  - as we also use Data2Vec2 as the teacher in many experiments, this allows us a close comparison between the models, as they are trained on the same data
  - data augmentation includes random resized crop, followed by a random horizontal flip, followed by normalization each channel seperatly using channel wise mean and standard deviation computed from the imagenet training set -> standard procedure for image preprocessing, and used for all images throughout this work
  - random resized crop: crop a random part of the image, then resize it to the desired size (224x224)
    - crop size is hyperparameter, but we just use the same as in Data2Vec2, which is 0.08 to 1.0 of the original image size
    - 8% as lower bound seems to be a very low value, a lot of information is lost, but it is a common value in the literature, so we do

    the same
  - random horizontal flip: randomly flip the image horizontally (self-explanatory)
- for validation, we resize each image to the same size as in training (224x224) and normalize it using the same procedure as in training
- generally ALL images in this thesis will be scaled to the same size: 224x224
- we access the data from Huggingface's dataset hub

- benchmarks published on downstream tasks/dataset like GLUE, more on that later
- data is not that much and only meant for fine-tuning and benchmarking as downstream task
- as with images, we a just trying the replicate the outputs, i.e. the representations of the input data
- we can use any text (dataset(s)), as long as it is large enough
- for text data (pretraining, which is the Knowledge-Distillation that we do) we select openwebtext
  - dataset build to reproduce datasets used to train GPT-2

  - publicly available and popular, used by e.g. BEiT3
- we access the data from Huggingface's dataset hub
  - published as slices and without any split
  - we take subsets 0-4 for training and 5 for validation, which is about 25% of the data
- due to open source efforts, data is already preprocessed and cleaned
- we apply further preprocessing by removing empty lines and null bytes, which we found are quite common and lead to problems during encoding and training, as they provide no learnable information
- the text of every dataset, containing text, so also openwebtext, is tokenized and encoded using the GPT-2 byte-pair encoder (citation here -> same as in D2V), with a vocabulary size of 50262 tokens
- we seperate the sentences using the end-of-sentence token, also done by Data2Vec2
- also used by Data2Vec2, and we use it, again, for the purpose of comparison
- so save disk space, we save the training and validation sets of owt in a single binary file, respectively
  - the binary files already contain the encoded text, so that we only need to batch them during training
  - in order to ensure correct encoding and to save the time for implementing the binary encoding, we use the dataset functionality of Fairseq, a library for sequence-to-sequence models developed by Facebook/Meta, to encode an binarize the text data, which is also used by Data2Vec2
- in total, our openwebtext subset consists of more than 2.5 billion tokens and consumes roughly 6 GB of disk space
- really low, compared to the image data, which is about 150 GB
- we do not use bookcorpus and english wikipedia, datasets Data2Vec2 was trained on, as Openwebtext appears to be a more recent and popular dataset for Knowledge-Distillation -> DistilGPT2 and Distilroberta trained on openwebtext, results showed that this dataset yields good results for (knowledge) distillation
- multimodal data:
- we need to use datasets with image-text pairs
- as Data2Vec not multimodal, we do not have any reference datasets
- however, many multimodal models, like BEiT and VLMo use the same popular multimodal dataset
- we therefore also opt for them

- we use COCO and a subset of Google's Conceptual Captions
- even though COCO contains just contains 82783 images, which is not that much, it contains multiple captions per image, meaning we can create multiple image-text pairs from one image
- images have a little more than average 5 captions, yielding a total of 566747 actual examples for the training set (used for Knowledge-Distillation)
- we also additionally use SBU Captions (SBU) and a subset of Google's Conceptual Captions (CC3M), which originally contain 1M and over 3.3M unique image-text pairs, respectively
- COCO has comparatively few images, just 82,783, so we need more data to train the model
- also helps in balancing the ratio of unique images and text
- with just COCO, we have five times more variety in text than in images
- in SBU and CC3M each image is only associated with one caption, and because we use at least 10x more images than in COCO, we can reduce the relative overrepresentation of text
- both datsets do not provide the images directly, but instead an index with the url of an image, and a correspondig caption
- url point to various sources on the web, so there is no guarantee that all images are still available
- for SBU, we collect all image(-text pairs) that are available as of July 2024
- for Google's CC3m we use the first 800,000 available, as of July 2024, images (with their captions) as a subset from the training set index published by Google[1].
- storing the whole dataset is not feasible for us and the combination with COCO and SBU should already provide enough data for training, as shown in table Table 1
  - we are constrained by 600GB of disk space, which must also be shared with other datsets, e.g. ImageNet-1k
- ids and urls of image-text pairs used, for both SBU and CC3M, are available on GitHub < *TODO: Footnote with URL!!!* >

[1] https://ai.google.com/research/ConceptualCaptions/download

Dataset	# Images	Avg. Caption Length	# Image-Text Pairs
COCO [Lin+14]	82,783	11.0	566,747
SBU Captions [OKB11]	840,792	12.0	840,792
CC3M [Sha+18] (Subset)	1,516,133	10.3	1,516,133
Total	2,439,708	-	2,923,672

Table 1: Multimodal Dataset used for aligning Image and Text. The maximum text sequence length is, inspired by BEiT3, set to 64 tokens [Wan+23].

- aforementioned papers (e.g. BEiT-3) also use the popular Visual Genome (VG) dataset, containing 108,249 images with on average 50 captions per image
- we have that many captions, because the captions are actually region descriptions of the images, not a general/global description of the image [Kri+17]
  - can be very short and only capture a small part of the image
  - are used for describing regions/objects in the image
- still used by the papers
- we will not use it as we encountered problems when using it together with contrastive learning, which will be explained in section Section 2.2.1.2
- all captions are tokenized and encoded using the same GPT-2 byte-pair encoder as the text-only data
- as usual, and done in BEiT, VLMo, and FLAVA, we prepend each caption with a start-of-sequence token and append an end-of-sequence token
- we use the same data augmentation as in the unimodal case for the images
  - that is, during training we apply random resized crop, followed by a random horizontal flip, followed by the imagenet normalization
  - during validation, we resize the image to 224x224 and normalize it using the same procedure as in training
  - the only difference lies in the crop size of the images
  - min crop size set to 0.08 for image pretraining (unimodal image distillation)
  - means at a minimum we could crop 8% of the image, and discard the rest
  - destroys a lot of information, not a problem for image only training, but when the image has a caption describing the image, and some parts focusing on the cropped parts, which can especially happen for VG, where the captions consists of region descriptions, then we might have captions that do not match the image anymore
  - that is why papers that use random crop use higher values -> BEiT3 uses 0.5, FLAVA 0.9, VLMo uses RandAugment
  - we consider 0.9 too high and 0.5 too low, so we opt for 0.6
- examples of image-text pairs and the effect of the crop size can be seen in the appendix

- in order to evaluate the models performance on text-only tasks, we use the GLUE benchmark
- GLUE consists of 4 different tasks: sentiment analysis (SST-2), grammar error detection (CoLA), sentence similarity (STS-B, MRPC, QQP), and natural language understanding (QNLI, MNLI, RTE)

SST-2:
- sentence classification of rotten tomatoes movie reviews into "negative" (1), "somewhat negative" (2), "somewhat positive" (3), and "positive" (4) [Soc+13]

CoLA:
- grammatical error detection / linguistic accepability, binary classification, whether a sentence is grammatically correct (acceptable -> 2) or not (0 -> unacceptable) [WSB18]

STS-B:
- similarity of two sentences, regression task, similarity score between 0 and 5 [May21]

MRPC:
- paraphrase detection, whether two sentences describe the same content/concept, binary classification [DB05]

QQP:
- do two questions ask the same thing, binary classification[2]

QNLI:
- does a text contain the answer to a question? [Raj+16, Wan+19]

RTE:
- pair of text and hypothesis, whether the hypothesis can be inferred from the text, binary classification [Ben+09, DGM06, Gia+07, Bar+06, Wan+19]

MNLI:

- pair of premise and hypothesis, whether the hypothesis can be inferred from the premise (entailment), contradicts the premise (contradiction), or is neutral (neutral) [WNB18]
- single questions encoded using same GPT-2 byte-pair encoder as before, and prepended with start-of-sequence token and appended with end-of-sequence token
- sentences are tokenized (samte tokenizer), concatenated and separated by the end-of-sentence token, concatenated sentence pair prepended with start-of-sequence token

[2] https://quoradata.quora.com/First-Quora-Dataset-Release-Question-Pairs

Dataset	Example	Label
CoLA	Our friends won't buy this analysis, let alone the next one we propose.	1
SST-2	hide new secretions from the parental units	0
MRPC	Amrozi accused his brother, whom he called "the witness", of deliberately distorting his evidence. [SEP] Referring to him as only "the witness", Amrozi accused his brother of deliberately distorting his evidence.	1
STS-B	A plane is taking off. [SEP] An air plane is taking off.	5.0
QQP	How is the life of a math student? Could you describe your own experiences? [SEP] Which level of prepration is enough for the exam jlpt5?	0
MNLI	Conceptually cream skimming has two basic dimensions - product and geography. [SEP] Product and geography are what make cream skimming work.	1
QNLI	When did the third Digimon series begin? [SEP] Unlike the two seasons before it and most of the seasons that followed, Digimon Tamers takes a darker and more realistic approach to its story featuring Digimon who do not reincarnate after their deaths and more complex character development in the original Japanese.	1
RTE	No Weapons of Mass Destruction Found in Iraq Yet. [SEP] Weapons of Mass Destruction Found in Iraq.	1

Table 2: Glue Example

2.1.4 Contrastive Learning

- is used to compare samples (e.g. images) with each other in representation space, typically by some distance metric, of which cosine similarity is the usual choice
- used in self-supervised learning to learn representations without classical labels such as class targets
- goal is to learn (abstract) representation of the input modality, e.g. images
- originally used in computer vision
  - idea is, that a representation of one image should be similar, or very close, to augmented versions of the same image
  - after all, content of the image stays the same after augmentation (provided the augmentation is not too drastic, e.g. crop size too big)
- goal of the (image) model is to maximize the cosine similary between the original image and its augmented versions
- this alone not sufficient, as model will collapse to a trivial solution, by simply return the same representation for all inputs
  - will maximize the cosine similarity between between the original image and its augmented versions, as representation produced for an image will always be the same

- to prevent this, negative samples are introduced
  - negative samples are other images, so not the original image
  - (usually) does not contain the same content as the original image, so cosine similarity between the original image should be minimized
- that way, model can't collapse to a constant representation, as this would not minimize the cosine similarity, and thus not minimize the loss
- this can be extended from unimodal to multimodal applications, in our case: images and text
- here we would like to maximize the cosine similarity between an image and its corresponding text, i.e. caption, and vice versa
- we do not need any augmentation, as we always have pairs: one image and one text
- negative samples for images are captions of other images, and vice versa
- model learns to produce similar representations for an image and its caption, describing the same real-world concept

Implementation:
- staying at our multimodal case, contrastive learning/loss is usually done on the batch-level
- means the multimodal model creates representations for all images and captions in the batch
- then, the cosine similarity between, the representations, of all images and captions in the batch is computed
  - can be done efficiently by normalizing each embedding and then perform matrix multiplication
- for a batch size of e.g. 256, each image has 255 negative samples, i.e. captions of other images, and one positive sample, i.e. its own caption, and vice versa
- can be interpreted as a classification problem with 256 classes, where the model has to predict the correct class, i.e. the positive sample, out of 256 classes/representations
- result of matrix multiplication is a 256x256 matrix with logits, where the diagonal contains the cosine similarity between the positive samples, i.e. the correct class
- we can do softmax row-wise to get probabilities for each image, and column-wise to get probabilities for each caption
- cross-entropy can then be used as the loss function on the probability distributions, metric is accuracy

Problem:
- result highly dependend on the amount of negative samples that are available
  - as an example, if batch size would be two, then the model would have to differentiate between one caption that belongs to the image and one that does not (negative sample), and vice versa
  - a lot simpler than with 255 negative samples, or even more
- result will be better with more negative examples, as task more challenging

- more negative samples can be achieved by using larger batch sizes, but this usually require, depending on the model architecture, higher VRAM GPUs or even multiple GPUs
  - costly

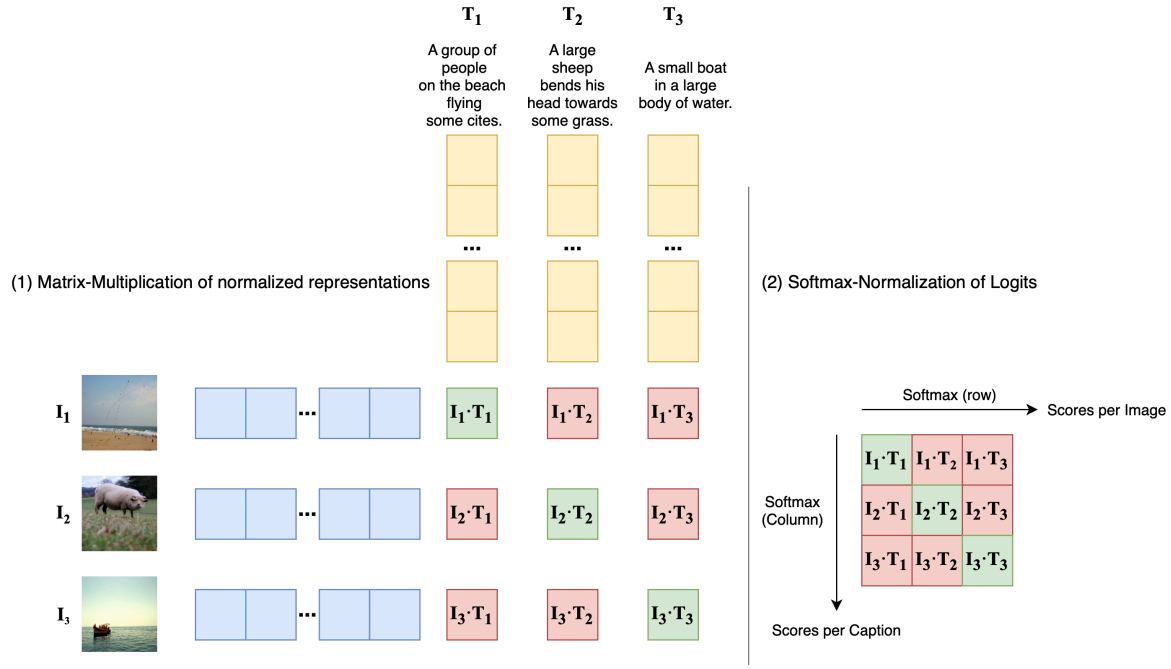

Figure 1: Contrastive Learning is performed using Matrix-Multiplication of normalized representations (1). The diagonal of the resulting matrix contains the cosine similarity between positive samples. The softmax operation along the rows yields a probabilty distribution for each image over all captions, and the softmax operation along the columns vice versa. The cross-entropy loss is then used to calculate the loss for the image scores and caption scores, respectively. The final loss is the mean of both losses. Image-Text pairs in the figure have been taken from the COCO train set [Lin+14].

2.1.5 Retrieval

- useful for benchmarking multimodal models
- cheap way, as is does not involve finetuning, just the embeddings produced by the model are needed
- goal: find matching (most similar) caption for a given image, and vice versa
  - means other part of the pair
- we first have a set of samples, e.g. images/captions, which we embed and normalize the embedding-> become a set of keys
- then we have a query, e.g. an image/text, which we also embed and normalize -> becomes a query
- now we compute cosine similarity between the query and all keys
- rank them based on the similarity, and retrieved sample is the one with the highest similarity

- computation can be done the same as described for contrastive learning Section 2.1.4 (contrastive learning and retrieval are basically the same)
  - only softmax operation is not needed (step 2 in Figure 1)
  - just take the maximum of all similarities -> most similar sample
- in application of this thesis, as contrastive learning, done for image-text pairs
- i.e. find caption for a given image in all captions of the dataset, that is used for benchmarking, and vice versa
- metric used is Rank@K (R@K), where K determines at which rank the paired sample has at least to be in the ranking in order to be considered as a correct retrieval
  - we use R@1, R@5, and R@10
  - R@1 is simply the normal accuracy, i.e. paired sample has to be the most similar one
  - R@5 means that the paired sample has to be in the top 5 most similar samples
  - R@10 means that the paired sample has to be in the top 10 most similar samples
- in this thesis we use the 5k test set of MSCOCO [Lin+14], and the 1k test set of Flickr30k [You+14] for benchmarking
- used by most multimodal models like FLAVA [Sin+21], CLIP [Rad+21], VLMo [Bao+22], and BEiT-3 [Wan+23]

-> we can easily and cheaply compare our model to those papers/models

2.1.6 Differences to Unimodal Knowledge Distillation

- for multimodal Knowledge Distillation, we also need a teacher model
- question is, which model should be the teacher model, or rather, in which modality should the teacher have been (pre-)trained?
- in our case, should the teacher be an image model or a text model?
- why not both?, so why not have a teacher text model and a teacher image model, so two teachers?
- because then the model would have to regress two representations, one for the image and one for the text
- would mean for an image-text pair, the student would regress two representations, one for the image and one for the text
- because both teachers are not related, the representations of the image and text, we would like to learn, are not aligned and related in any way
- recall that a multimodal model always has at least one shared block at the end of the architecture
- constrains the model that the same representation has to be produced for an image and its corresponding text/caption
- only then we can align the representations of the image and text
- if we would now have two teachers, one for the image and one for the text, and the student would regress both representations, then we would have two targets for one image-text

pair, but we can only predict one representation, which should be the same for the image and text
- also, with two targets the model would have to learn two different representations for the image and text, and would most likely not learn

anything meaningful, as it is not possible, and not desired, to learn two different representations at the same time, i.e for an image and its corresponding text
- so we only can take one teacher model, either an image model or a text model
- we select an image model as the teacher model
- also done by [AVT17], which pioneered the approach
- seems to work good in practice, so we follow this approach
- makes sense, as there are a lot of supervised, and self-supvervised, image models available, and learning content from image

well researched

- also: VLMo initializes attention layers, which are shared between image and text, with weights from a pretrained image model [Bao+22]
- so it seems a model can learn text based on knowledge obtained from image pretraining
- consequently, we will use, as in unimodal distillation, a pretrained image model as the teacher model
  - so we still have just one teacher model
  - also good, because a second one would increase the computational cost, GPU memory requirements, and training time
- but why not directly use a multimodal model as the teacher model?
- because the goal is to learn a multimodal model (learn alignment of modalities, i.e. image and text) from scratch
- and use the fact that there are many pretrained unimodal (for us now image) models available
- recall: goal of this research/thesis is to learn it from just a unimodal teacher model!!!
- previously, in unimodal knowledge distillation, we were able to regress all time steps of the teacher model with the student model
- means the representation of each patch or text token, respectively
  - included the CLS/BOS token
- for multimodal Knowledge Distillation, we can't do this
  - we have to regress the whole image/text representation
  - and not the representation of each patch or text token
- has two reasons

1.

- the number of time steps (patches) an image model has is usually not the same as the number of time steps (text tokens) a text model has
- so we can't just regress all time steps
- also, text can vary in length, and we use padding -> embedding at the time steps where there is padding is not meaningful
- we would regress the representation of individual image patches -> if an image time step (patch) contains e.g. an eye, and the text token at the same time step is a padding token, then regressing this does not make sense

2.

- in order for this to work, a text token at a certain time step has to be related to the image patch at the same time step
- so if an image patch contains an eye, the text token at the same time step has to contain the word "eye"
- not possible, as result would be just a concatenation of words and no meaningful text
- also, text naturally contains fill words, e.g. "the", "a", "is", which is nothing that can be represented in any way in an image
- also those words do not have any meaning regarding real-world concepts, like a dog, cat, or car
- example illustrated in Figure 2

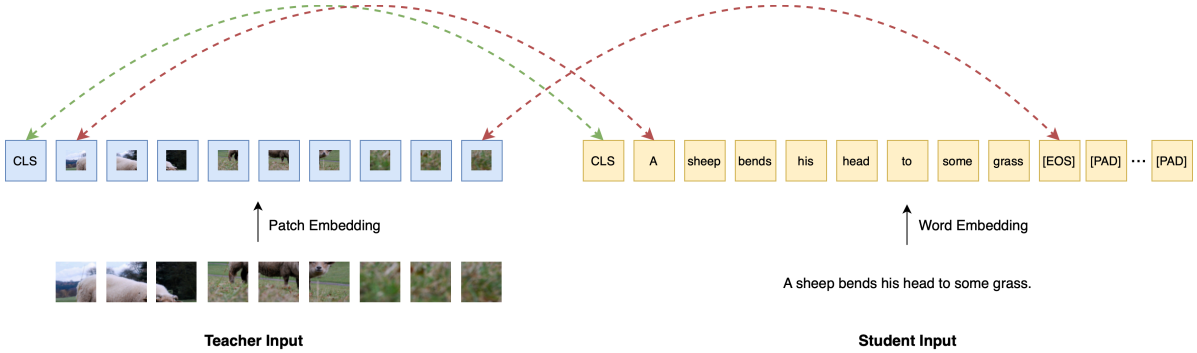

Figure 2: The meaning/content of time steps across modalities is not aligned, and the number of time steps will differ between modalities. This makes alignment on the level of individual time steps impossible. The CLS token aggregates global information independent of time steps, and captures the meaning and interpretation of the respective input, making alignment possible. This requires the teacher CLS token to not contain any modality-specific (in this case image) information after the last layer. Image-Text example is taken from the COCO train set [Lin+14].

- that is why we have to regress the global representation of the image and text
- means the CLS/BOS token -> goal of it is to aggregate as much information as possible, meaning it is a global representation of the image/text content

- is independent of the number of time steps (patches) or text tokens or what is going on in a certain time step
- necessary requirement: representation of CLS token that is returned by the teacher and regressed by the student has to be independent of the image modality
- means it should be abstract enought that it can be used to also describe the content of the caption of the image
- if the representation of the CLS token still contains image specific information, then the student model will not be able to align the representation of the caption with that of the image
  - based on the caption, it is impossible to predict the image-specific information still encoded in the representation of the CLS token
  - also not desired, representation should be independent of the modality
- this is something we will elaborate on in < TODO: cite >
- SHRe can be seen as a special case of regressing the CLS token [AVT17]
- was published before the inception of Transformers [Vas+17]
- uses ConvNets
- output of FFN head of deep ConvNets usually contains global information of the image due to the increased receptive field with more layers
- so in a sense, SHRe does exactly what we aim to do, just in a supervised way: it regresses the probability distribution of the Imagenet-1k classes
- probability distribution created from FFN head of the ConvNet, contains global information of the image, like the CLS token in Transformers

2.2 Multimodal Knowledge Distillation

2.2.1 Aligned Representations

2.2.1.1 Dual encoder

2.2.1.1.1 Unimodal Student

- we start as simple as possible
- we just want to create aligned cross-modal representations, i.e. have the same representation for the same concept -> same representation for an image and its corresponding text (image-text pair in the dataset)
- multiple architectures are possible, we start with a CLIP-like architecture
- means one encoder per modalitiy, so one for image and one for text

- usually trained for scratch -> but expensive, and we want to utilize already pretrained(!) (self-supervised, not trained on labeled data) unimodal models
- we use BEiT-2 as the teacher model, which is an image model, and at the same time serves as our image encoder
- to now align representations, we just train the text encoder to regress the cls token of the image encoder
- hope is that the cls token of the teacher, which is regressed, has learned a representation that is abstract enough so that it can be applied independently of the modality, in this case images
  - although there has been no incentive for the teacher model to learn which is independent of the modality, as the model has only been pretrained on images -> if the representation (cls token) does still contain image specific information, the student model will not be able to learn a meaningful representation of the text, as image and text are inherently different
- in the first experiment we just train the text encoder to regress the cls token of the image encoder, nothing more
- the text model is smaller than the image model, it contains only 7 layers, while the image model contains 12 layers
- hope is that through Knowledge-Distillation we do not need a model as large as the teacher model, as we have seen that a smaller model can achive a performance quite similar to the larger teacher model through KD in the unimodal case
  - whether this translates to an multimodal setting can be derived from the results of this experiment
  - this could be seen in the retrieval application of our model -> if the performance on text-image retrieval is similar to the performance of the teacher model, then we can assume that the student model has learned a meaningful representation of the text(?)
- we still have the option to expand our student model to the same size as the teacher model, i.e. 12 layers
- still relatively cheap, as we only have to train the text encoder and, as we are doing in the first experiment with 7 layers, we can initialize the text encoder (the student) with the weights of the text D2V2 model, meaning we do not start from scratch

2.2.1.1.2 Reproducing: “See, Hear, and Read: Deep Aligned Representations” [AVT17]

- we start as simple as possible
- we first want to reproduce the results of the paper “See, Hear, and Read: Deep Aligned Representations” [AVT17]
- for the sake of simplicity, we will refer to the paper as SHRe (SEe, HEar, and REad), as the title is quite long and the authors do not name the architecture
- will give us a baseline on which to improve, compare our results to (especially because SHRe does not benchmark retrieval on karpathy COCO), and test new ideas

- we use the same architecture as in the paper, which is a dual encoder architecture
  - has one text and one image encoder/network, quite like CLIP does
- difference to CLIP is: we now have a shared network at the top
- for our image and text encoder we use the pretrained image D2V2, and the pretrained text D2V2, model respectively
  - this differs to SHRe, as they used conv nets for both image and text encoder
- unlike SHRe, we directly initialize the encoders with the weights of the respective D2V2 model, and do not train them from scratch
- further, because we would like to keep the model smaller, which is less expensive to train and we can train it, thanks to Deepspeed, on a single GPU, we only use the first 7 layers of the D2V2 models
- SHRe added two MLP layers as the shared network, we will do the same, but use the same MLP architecture as present in Transformer layers
  - means, two linear layers, with a GELU activation and layer norm in between
  - first linear layer expands the embedding dimension to 4 times the size, second linear layer reduces it to the number of classes in imagenet (1000), not the embedding dimension
- recall the SHRe uses the kl-divergence loss, common in labeled KD
  - means the output of our multimodal model, i.e. shared network, is a probablity distribution over the classes in imagenet
- we regress the probablity distribution of a resnet-b-50 model, which was trained on imagenet with labeled data, and is used as the teacher model
  - SHRe did not mention which architecture they used, only that teacher was trained on imagenet
- to pass the features from the modality specific encoders (image and text), which operate on time steps, to the shared network, which does not operate on time steps, we use the cls tokens of the output of the encoders (the first token)
- SHRe uses two training targets: minimization of the KL-divergence, and the maximization of the cosine similarity for the activations of the shared network between matching image-text pairs (positive) and the minimization of the cosine similarity for the activations of the shared network between non-matching image-text pairs (negative)
- this is basically a contrastive loss, similar to that of CLIP
- we do not exactly follow this approach, but instead the of VLMo
- we compute the cosine similarity between the normalized logits of the shared network for all possible image-text pairs of a batch
  - we take the output of our model, i.e. the shared network, without softmax, and normalize it
- and then take softmax over the cosine similarities, and then compute the cross-entropy loss
- for each image/text, the target is the matching text/image in the batch, of which there is only one, and the rest texts/images in the batch are non-matching

- following CLIP, we divide the cosine similarity by a, in log-space, learnable temperature parameter
- this way we align image-text pairs in predictions, i.e. class probabilities, and in represenation space

Training Setup:

- we train the model for 60k steps, with a batch size of 256
- we train using Deepspeed stage 2, to be able to train on a single GPU
- we use the AdamW optimizer, with a peak learning rate of 5e-4
  - we do not have the resources for extensive tuning of the learning rate, and other hyperparameters
  - we therfore use the same learning rate as in the unimodal experiments, which gave good results
- we set the AdamW epsilon to 1e-6, weight decay to 0.01, and the betas to (0.9,0.98)
- we use cosine scheduling, with a warmup of 10% of the total steps -> 6k steps
- as explained in the chapter about the datasets, we set the max text sequence length to 64 tokens
  - captions tend to be small and concise, as shown in section < TODO >
  - enables faster training and less memory usage
- we use COCO and Conceptual Captions in a round-robin fashion
- text encoder and image encoder are initialized with the weights of the respective D2V2 models
  - we only use the first 7 layers of the D2V2 models so that the model is smaller
  - therefore, each encoder has 7 transformer layers
- shared network is a two-layer MLP, and follows the same architecture as the MLP in a standard transformer layer
  - first layer has 4 times the size of the embedding dimension, second layer directly reduces it to the number of classes in imagenet (1000)
  - second layer is output layer, initialized with random weights
- we use the CLIP implementation of imagenet zero-shot classification during validation, which we run every 6k steps

Type	Hyperparameter	Value
Image/Text Encoder	Layers	7
	Hidden size	768
	Attention heads	12
	FFN inner hidden size	3072
	Shared Encoder FFN output size	1000
Training	Training steps	60,000
	Batch size	256
	AdamW ε	1e-6
	AdamW β	(0.9, 0.98)
	Peak learning rate	5e-4
	Learning rate schedule	Cosine
	Warmup steps	6k (10%)
	Weight decay	0.01
	Teacher	Imagenet ResNet-50

Table 3: Hyperparameters of reproduced SHRe model.

After the training, we evaluate the model on the same CLIP zero-shot imagenet classification task, and using image-text retrieval on the MSCOCO and Flickr30K datasets. In both case the model is relieant on the quality of the representations it produces, making it a good benchmark. Further, we can compare the results directly to other papers like FLAVA, VLMo, and CLIP, and do not need to do any seperate finetuning. Both benchmarks are a direct indicator of the success of the method.

Model	Accuracy
Random	0.001
Visual N-Grams [Li+17]	11.5
CLIP [Rad+21]	**72.6**
SHRe reproduction (ours)	21.8

Table 4: First comparison of zero-shot imagenet classification accuracy with CLIP and Visual N-Grams.

As seen in table Table 4, the model achieves almost double the accuracy as Visual N-Grams, which was the initial approach on zero-shot transfer for classification. However, CLIP outperforms our model by a margin. This is to be expected as:
1. The reported accuracy uses a model setup with 428 million parameters.
2. The model was trained on up to 400 million image-text pairs.
3. The model was trained on 256 V100 GPUs for 12 days.

Huge difference to our model, which has 144 million parameters, was trained on just short of 1.4 million image-text pairs, and was trained on a single RTX 4090 for 7 hours. The cost accounts to 5.25 USD, compared to more than 73,000 USD for the CLIP model[3].

- table Table 5 shows results of image-text retrieval on MSCOCO and Flickr30K
- for now, we only compare to CLIP and FLAVA, as this is the only dual encoder model, i.e. the image and text encoder are completely seperate
- we will compare to VLMo and BEiT-3 in the section on Mixture-of-Modality-Experts
- we observe that model performs quite well on MSCOCO, but not so well on Flickr30K
- reason might be that MSCOCO training and test datasets are very similar, and since we only use Conceptual Captions as additional data, the model might be very biased towards MSCOCO
- would explain poorer performance on Flickr30K, as the model has not seen any Flickr30k data during training
- we can even see that the model performs slightly better image retrieval for MSCOCO on R@10, and is, for the model size and training data, quite close to CLIP and FLAVA
- important to note that the teacher is still supervised, which is a clear advantage over CLIP and FLAVA, which are trained from scratch, without KD
- will be even more interesting to see how the model performs when the teacher is self-supervised, and does not provide us with a probability distribution to regress -> later sections

Model	**MSCOCO (5K test set)**						**Flickr30K (1K test set)**					
	Image → Text			Text → Image			Image → Text			Text → Image		
	R@1	R@5	R@10	R@1	R@5	R@10	R@1	R@5	R@10	R@1	R@5	R@10
Zero-Shot												
FLAVA [Sin+21]	42.74	76.76	-	38.38	67.47	-	67.7	94.0	-	65.22	89.38	-
CLIP [Rad+21]	58.4	81.5	88.1	37.8	62.4	72.2	88.0	98.7	99.4	68.7	90.6	95.2
SHRe reproduction (ours)	41.36	71.16	82.0	30.2	59.46	72.54	9.5	35.68	50.18	8.38	37.54	49.88

Table 5:

These results can already be considered as a success, as the aim of this work is not to reach state-of-the-art performance, but to create a poof-of-concept for multimodal knowledge distillation, although a high performance is desirable.

[3]Calculation was done based on the hourly GPU cost on runpod.io, which is the platform we used to rent GPUs. As of the time of our research, a single RTX 4090 costs 0.75 USD, and a single V100 1 USD per hour.

2.2.1.2 Region Descriptions with Contrastive Learning

- many papers use the Visual Genome dataset, consisting of images with region descriptions -> attractive source, as region descriptions are human annotated and highly curated -> focus on specfic regions of the image
- as mentioned in section Section 2.1.3, we do not use Visual Genome because we encoutered problems when using it with Contrastive Learning
- Figure 3 shows accuracy on image-text contrast, which is image-text retrieval, when using data datasets in combination with Visual Genome and without

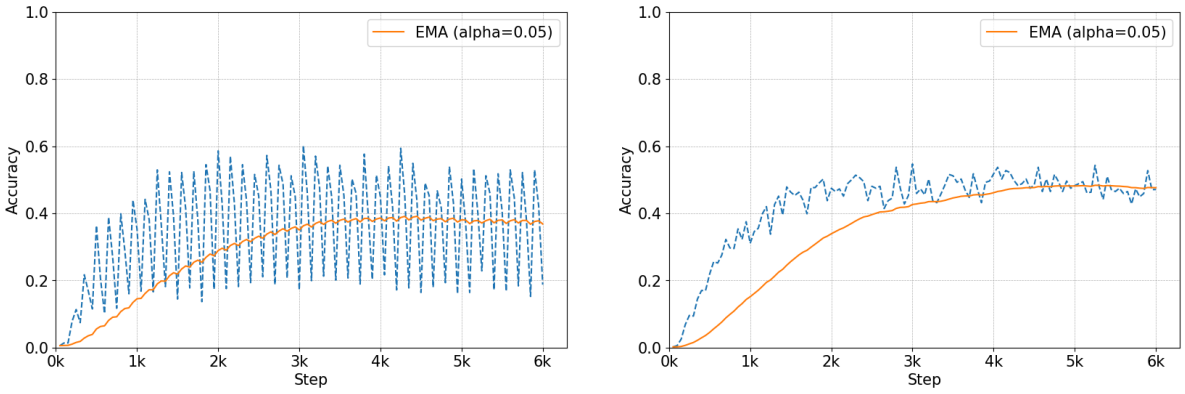

Figure 3: Training accuracy of Image-Text Contrast with Visual Genome (left) vs. without Visual Genome (right). Removing Visual Genome from the training data leads to a more stable training and a higher accuracy in the first 6k steps.

- comparison is only for the first 6k steps, as we stop experiments that show errors or do not seem promising, due to the high computational cost
- contrary to expectations, the accuracy of the model without Visual Genome continously increases after the first 6k steps (Figure 4), where is stagnates for a while

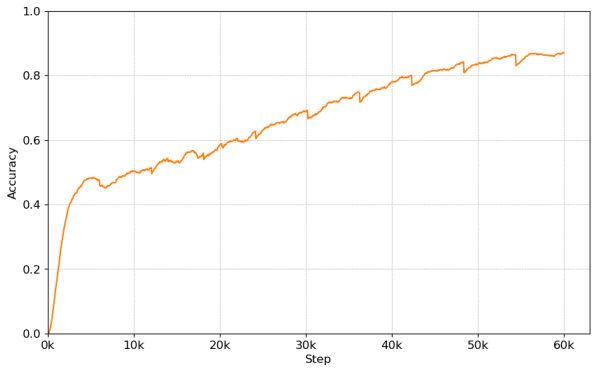

Figure 4:

- we assume reason is that region descriptions are too specific, i.e. focus on a specific part/ region of the image, and do not capture the overall content of the image
- also, since the regions can be small, the caption will also be

2.2.1.3 On FLAVA's retrieval performance

- FLAVA authors claim their performance on image-text retrieval on MSCOCO and Flickr30K is zero-shot
- however, this is not true
- they train there model using contrastive loss, among other losses
- as mentioned in Section 2.1.5, contrastive learning and retrieval are basically the same
  - contrastive learning uses cosine similarity to push embeddings of matching pairs closer together, and embeddings of non-matching pairs further apart
  - retrieval uses cosine similarity to find the matching pair for a given query (aims to e.g. find the caption for a given image)
- so with contrastive learning, the model learns to perform retrieval
- zero-shot is when a trained model is applied to a task it has not been trained on, e.g. when a multimodal model is trained only using masked-image-modeling (MIM) and masked-language-modeling (MLM), and then applied to image-text retrieval
  - or when an LLM is applied to a tabular classification task
- however, FLAVA uses contrastive learning, and therefore the model has been trained on the retrieval task, and is not zero-shot
- also the model has been trained using the MSCOCO train set, and the image-text retrieval is done on the MSCOCO test set
- if the samples from MSCOCO train and test set are similar, then it is just a normal application of a trained model to a task it has been trained on (among other tasks)
- application of itr on Flickr30K is still not zero-shot, as, again, the model has been trained using a contrastive loss
- one example for actual zero-shot retrieval would be BEiT-3, having only been trained using MIM and MLM [Wan+23]
  - finetuning BEiT-3 on Flickr30K using contrastive less then show slight improvement

Model	**Flickr30K (1K test set)**					
	Image → Text			Text → Image		
	R@1	R@5	R@10	R@1	R@5	R@10
BEiT-3 *zero-shot*	94.9	99.9	**100.0**	81.5	95.6	97.8
BEiT-3 *finetuning*	**98.0**	**100.0**	**100.0**	**90.3**	**98.7**	**99.5**

Table 6:

2.2.1.4 Increasing Negative Samples

2.2.1.4.1 Memory Bank

- as mentioned in the section about contrastive learning Section 2.1.4, quality of representations learning using contrastive loss greatly improves with more negative samples

-> for example, CLIP, trained only using contrastive learning, uses a batch size of 32k [Rad+21], the the large variant of VLMo between 16k and 32k [Bao+22], and FLAVA 8k [Sin+21]

- not necessary though -> base model of VLMo uses "just" 1024 [Bao+22], and SHRe just 200 [AVT17], both achieve, as described in the respective chapters, good results
- we are limited by GPU memory, currently using a batch size of 256, can't increase it without further optimizations or multiple GPUs

2.2.1.4.2 Larger Batch Sizes with GPU Offloading

2.2.1.4.3 Feature-based Knowledge Distillation

- if we want to keep the approach from SHRe, i.e. use KD and contrastive learning, for a self-supervised trained teacher that does not provide us with a probabilty distribution to regress, then we have to regress the activations/features of the teacher model
- this is similar to the unimodal distillation in the first experimental sections
- however, we can't just regress all time steps of the teacher model, i.e. all activations of the teacher, as we did in the unimodal case
  - this works for the image encoder of our multimodal model, because the teacher model is an image model, but not for the text encoder
- firstly, is is not possible because the text might not have as many tokens (time steps) as the image has patches (time steps)
- so regressing a timestep/patch of the image, say 180, with the same timestep of the text, also 180, is not possible if the text is not that long
  - in our case for example, the text has a max sequence length of only 64 tokens
- even if we were to set the max text sequence length to the same number of time steps as an image has (196+1), i.e. through padding, it will certainly never be the case that a specific timestep/patch of an image aligns semantically with the text token at the same timestep
- the reason is that each time step for an image corresponds to a patch, which contains some information
- if we want to align the representation of that specific patch with a text token as the same timestep, then the text token has to contain the same information as the patch in text form
  - e.g. if a patch contains a piece of cheese, then the text token at the same timestep has to contain the word "cheese"
  - this would have to hold for all timesteps
  - so no fill words or padding allowed, as they do not contain any information
  - text would not make any sense and would not describe the overall content of the image
- therefore, we have to somehow regress the global information of the image, and can't regress any specific patch

- this is where the cls token comes in handy, as its goal is to aggregate a global representation of the image
- this is independent of any patch
- if our text encoder also has a cls token, then we can make the cls token of the text encoder regress the cls token of the (image) teacher
- in principle, for the image encoder we could regress all timesteps as in the unimodal case, as this is the same modalitiy, but we should keep it consistent and regress the cls token of the image encoder as well
- in principle nothing "bad", as probabilty distribution originates from a linear layer, which uses the cls token of the last layer
- also: makes the encoders push as many information as possible to their cls token
- to first validate if regressing just the cls token of the teacher model even works with a teacher that has been trained with labels, we first use the exact same approach as before, i.e. keep the supervised teacher model, but regress the cls token of the teacher model now
- we do not need any linear classifier on top of our model, and we also do not need it from the teacher model
- we just take the cls token output of the last transformer layer of the teacher model as the target
- questions is whether the information contained in the cls token of the teacher, which has now been trained without lables, are abstract enought...

2.2.1.4.4 Modality-invariant Representations

- we want to learn a representation that is invariant/independent to the modality
  - optimally representation of an image should not change if we pass a caption (text) of the image through the model
- can be learned e.g. by contrastive learning, here image-text contrastive learning
- we are not only using contrastive learning, but also KD to align the representations of the image and text
- we regress the output of a teacher model, in the above case the probability distribution over the 1000 classes of imagenet
- because this forces the model to learn a representation which is more focused on real-world concepts/objects, like e.g. a cat, the focus is shifted away from the modality specific features
- this is actually what we want -> we do not want to regress any representation of probabilty distribution that still contains modality specific features/information

    - imagenet classes are not modality specific, therefore the probabilty distribution we regress does not contain modality specific information
        - a cat is a cat, no matter if we see it or read about it
- this is why the whole process works -> we can still regress the probabilty distribution of imagenet, even though we are processing text
    - e.g. a text about a cat should have the imagenet class “cat” as the highest probability
- generally, there has been no incentive for our unimodal teacher model to learn a representation that is independent of the modality
- however, labels are modality independent, so the ouput of the model, i.e. the logits become modality independent
- goal of this paper is to create a multimodal model from an unimodal one, with the constraint that the unimodal (teacher) model has not been trained on labeled data -> we do not want to rely on labeled data -> self-supervised
- the question is, if the model is unimodal, meaning there has been no incentive to learn a representation that is independent of the modality, and there were no labels involved that could push the model to learn representations independent of the modality, then can we still learn a representation that is modalitiy-invariant?
- this is crucial, as if the teacher image model, e.g. BEiT-2, outputs a cls token that still encodes image specific information, then the student text model will not be able to learn a meaningful representation of the text
- so what it boils down to is: Do unimodal, self-supervised trained, models learn representations abstract enough so that they can be regressed by models of a different modality?

2.2.1.5 Contrastive Learning with Memory Bank

- currently we do contrastive learning based on all samples of the current batch
    - for an image, its corresponding text is the positive example, all other 255 (we use batch size of 256) captions are negative examples
- generally, it is advised to use bigger batch sizes with contrastive learning -> increases number of negative examples
    - VLMo uses batch size of up to 32k, FLAVA 8192, CLIP also 32k
    - comparison between VLMo models trained with batch size of 1024 and 32k showed that this improves the model greatly
    - FLAVA mentioned it made their training more stable
- not feasible for us -> max batch size, with deepspeed stage 2, is 256, gpu memory is full
- we would still like to have more negative examples, while keeping batch size the same
- solution is memory bank
- Initially popularized by [Wu+18]
- initially developed for self-supervised learning on images only

- stores the representations/embeddings of all samples of the dataset, this is the memory bank
- for each sample passed through the model, i.e. each sample in the batch, embedding is computed and cosine similarity to all embeddings in the memory bank, i.e. all samples of the dataset, is computed
  - softmax is then applied over the cosine similarities -> each image in the dataset, i.e. each sample in the memory bank, is seen as one class
  - for n images in the dataset, we have an n-class classification problem
- each time an image is padded through the model, the embedding is updated in the memory bank
- helps to increase the number of negative examples, without increasing the batch size

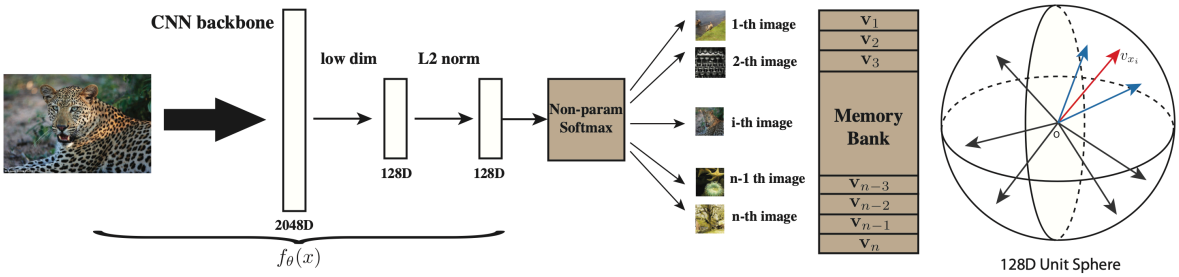

Figure 5: Memory Bank

- disadvantage:
  - memory bank is very large for large datasets, e.g. ImageNet -> they report 600MB for 128 dimensional embeddings, we use 768
  - too much memory for our GPU setup
  - classification problem -> as in our contrastive loss, we do softmax over all samples/classes
    - will take long on large datasets
  - embeddings are only updated when their sample is passed through the model
    - for large datasets, some embeddings might be old and not of high quality
- we use a memory bank, but not based on the size of the whole dataset -> we want to "simulate" the contrastive loss of large batch sizes
- means memory bank of size similar to the batch size of e.g. VLMo -> 16k-32k
- should fit on GPU
- softmax over less classes, so relatively fast -> should not be the bottleneck operation
- not all samples of the dataset will be in the memory bank
- for each batch, we do contrastive loss over the batch and all samples in the memory bank
- we treat the memory bank as a FIFO queue

    - after contrastive loss has been computed for current batch, we add it to the memory bank and discard the oldest batch
- depending on the size of the memory bank, samples in the memory bank are more "fresh"
- during start of the training, we progressively fill the memory bank with batches
- during the first iterations/batches, memory bank will not be full
- we therefore only do the contrastive loss based on the current amount of batches in the memory bank
- after some steps, depending how large the memory bank is, the memory bank will be full, and we do contrastive loss with the whole memory bank
- from this point on, with each batch we will replace the oldest batch in the memory bank with the current batch
- we need two memory banks, one for images and one for text

Init results:
- for 16k -> model bad -> task prob too difficult
- for 1024 -> model starts well, then collapses -> lr too high (5e-4, but much lower in other itc papers)

2.2.1.6 Scaling Memory Bank

2.2.1.7 Feature Whitening
- as noted by @feature_whitening, masked image modeling (MIM) usually ahead of contrastive learning
    - especially for downstream/finetuning tasks
- authors report increased performance, when distilling contrastive models using a special feature distillation, on downstream tasks
- fits our use case, as we already do distillation and contrastive learning
- higher performance on downstream task, like imagenet (zero-shot) classification is desirable -> as mentioned by FLAVA authors, multimodal model should not only perform well on multimodel tasks, but also on unimodal tasks

...
- they use feature map as targets, not logits, as some models do not have logits, i.e. probability distributions, as output/target to regress
- exactly what we are aiming for, as self-supervised models do not have logits as outputs, but feature (maps)

2.2.1.7.1 Adding Image-Text Contrast

- until now we did not actually used the same philosophy as in CLIP, which relies on, next to a seperate image and text encoder, a contrastive loss to align the representations of the image and text, so does not do KD and trains both text and image encoder from scratch
- as mentioned in the chapter about CLIP, the architecture features two linear projections, one for each modality/encoder
- goal is to project the image/text represenation in a shared multimodal embedding/latent space, on which the contrastive loss is computed
- if we also manage to do this successfully, the performance on image-text retrieval should increase by a margin

2.2.1.7.2 Stagewise Unimodal Distillation

2.3 Seperate Self-Attention

2.3.1 Baseline

- currently only 6 layers, 5 out of which are modality specific, 1 is shared
- we experiment with adding one additional moadality specific layer, and one additional shared layer in another experiment

-> more difficult to align mutliple modalities, than just training one -> add one layer -> motivation for modality specific: after 5 layers information might not be high level enough so that one layer can process the information -> add one additional modality specific -> motivation for shared: after 5 layers information might be high level enough, but capturing modality agnostic information might take more than one layer -> add one additional shared

- added shared layer improves performance slightly, but adds 7 million parameters and 41 minutes to training time
- looking at the improvement in zero-shot, which increases the average Recall from 29.93% to 30.8%, this is not much of an improvement, considering the amount of parameters we add to the model

Model	MSCOCO (5K test set)						Flickr30K (1K test set)					
	Image → Text			Text → Image			Image → Text			Text → Image		
	R@1	R@5	R@10	R@1	R@5	R@10	R@1	R@5	R@10	R@1	R@5	R@10
FLAVA	42.74	76.76	-	38.38	67.47	-	67.7	94.0	-	65.22	89.38	-
Data2Vec2	0.02	0.08	0.22	0.01	0.10	0.19	0.02	0.12	0.26	0.02	0.06	0.12
MM-D2V2 (Ours)	4.24	12.12	17.96	1.77	6.54	10.91	1.2	4.88	8.18	0.54	2.52	4.58
MM-D2V2 (Ours)†	31.72	56.78	67.9	12.42	31.05	42.5	7.7	26.18	37.6	4.08	17.01	24.26
MM-D2V2 7_2(Ours)†	32.78	58.34	69.3	12.83	31.85	43.4	8.08	27.92	38.6	4.14	17.5	24.82
MM-D2V2 7(Ours)†	30.24	56.48	67.46	11.96	30.48	41.88	7.36	26.42	36.6	3.7	16.58	23.84

Table 7: Comparison of Zero-shot Image-Text and Text-Image Retrieval of first results with FLAVA and Data2Vec2 papers. Because Data2Vec2 is a unimodal model, we embed each image with the D2V2-Image model and each text with the D2V2-Text model. This yields unusable results, as there has been no incentive for the models to learn a shared representation, as both are unimodal. This is why we had to use both the image and the text model to embed the data.

†: This version has been trained with BEiT-2 as the teacher model, not the D2V2 Image model.

Model	MSCOCO (5K test set)						Flickr30K (1K test set)					
	Image → Text			Text → Image			Image → Text			Text → Image		
	R@1	R@5	R@10	R@1	R@5	R@10	R@1	R@5	R@10	R@1	R@5	R@10
Zero-Shot												
FLAVA	42.74	76.76	-	38.38	67.47	-	67.7	94.0	-	65.22	89.38	-
CLIP	58.4	81.5	88.1	37.8	62.4	72.2	88.0	98.7	99.4	68.7	90.6	95.2
MM-D2V2 (Ours)	31.72	56.78	67.9	12.42	31.05	42.5	7.7	26.18	37.6	4.08	17.01	24.26
Finetune												
BEiT-3	84.8	96.5	98.3	67.2	87.7	92.8	98	100	100	90.3	98.7	99.5
VLMo	74.8	93.1	96.9	57.2	82.6	89.8	92.3	99.4	99.9	79.3	95.7	97.8

Table 8:

- looking at the validation loss of image and text seperatly, on COCO val set, we observe that the loss on images is significantly lower than the loss on text, which might be due to the fact that the teacher model is a vision model and the target, the cls token, might be biased towards the image modality, as it is unimodal
- interestingly, this bias also seem to be directly translated to the performance on image-text retrieval, as the performance on image-text retrieval is significantly higher than on text-image retrieval -> we are learning the cls token representation, and using the leared

cls token as an output for the student model, to encode a modality for retrieval and other downstream task

-> suggests that the cls token is biased towards the image modality, or rather that the model is better in encoding images than text

- we can see that the performance of e.g. BEiT-3 and VLMo is also lower on text-image retrieval than on image-text retrieval, but not the the extend that we observe with our model

Model	MSCOCO (5K test set)		Flickr30K (1K test set)	
	Image → Text	Text → Image	Image → Text	Text → Image
MM-D2V2 7(Ours)†	51.39	28.11	23.46	14.71
BEiT-3	93.2	82.57	99.33	96.17
VLMo	88.27	76.53	97.2	90.93

Table 9: Average recall of image-text and text-image retrieval on MSCOCO and Flickr30K. All models continously perform better on image-text retrieval than on text-image retrieval, but the difference is more pronounced for our model.

- currently vl layer(s) (or rather mulimodal layer(s)) are randomly initialized, one option is to specifically initialize the multimodal layers with the weight of the final layers of the D2V text model -> initial state is closer closer to text modality -> did not work
- currently embedding layer of text and patch embed layer of image model frozen, includes cls token of text and image frozen
- fuzed layer takes the cls token of the image model and the cls token of the text model as input
- so maybe unfreezing the embedding layers will help

-> did not work

- however, when we disable weight decay for image cls token and for text embedding layer, contains text cls/bos token, then we observed an increase in performance
- for all other params, weight decay stays at 0.01
- test adding image-mm and text-mm projections between encoders and shared encoder (FLAVA)
  - check retrieval performance when using ourputs of encoders, are they already a little bit aligned?
  - check retrieval performance when using outputs of projections, are they more aligned?
  - check retrieval performance when using outputs of shared encoder, are they even more aligned?
  - avg. cosine similarity for positive pairs and negative pairs

- we do not compare with See, Hear, and Read: Deep Aligned Representations use the average median rank instead of recall at a specific percent, and from their experimental setup it is not clear which samples they used from Visual Genome for their retrieval experiments.
- what is interesting however, is that with just model transfer, which is Knowledge-Distillation in our case, their model did not perform well on zero-shot retrieval -> halfed score of linear regression -> especially for image-sound retrieval just model transfer, i.e. labeled variant of KD, did not work well
- what made the important difference, which might also be the case for us, is Contrastive Learning, which, with the exception of BEiT-3, was used by both VLMo, FLAVA as one of the pretraining tasks, and for CLIP it was the only pretraining task

2.3.2 Image-Text Contrastive Learning

- solution is to also incorporate contrastive learning into our training
- as we still do KD, we now have two losses, the KD loss and the contrastive loss
  - nothing unusual, done by VLMo, FLAVA (use masked modality modeling as second pretraining task)
  - only contrastive loss done by CLIP
- how is it done in the papers? -> generally always the same
  - take the cls token of the text encoder output, and the cls token of the image encoder output
  - project each of them into a shared embedding space
  - compute the cosine similarity between the image embeddings and the text embeddings of the current batch
  - projection is done by linear layer, popularized by CLIP -> done the same across VLMo, FLAVA, CLIP, BEiT-3
- VLMo additionally takes the output of cls token of the whole model, not of the text and image encoder, and projects it into the shared embedding space with different linear layers
  - each model usually has two projection layers, one for the image encoder and one for the text encoder
  - VLMo has four projection layers, two additional for the cls token of the VL-expert
    - one if the output of the VL-expert is for text, and one if it is for image
  - is surprising, as the the VL-expert forces to learn a shared representation, so projecting it into a shared space with seperate projection layers seems counterintuitive, more intuitive would be to use one projection layer for the output of the cls token of the VL-expert
  - authors did not provide a reason for this
- therefore, we will start with the following:

    - seperate projection layers for image encoder and text encoder -> used to project image/text into multimodal space (FLAVA)
    - one projection layer for the cls token of the shared layer(s) -> used for contrastive learning
    - for unimodal finetuning: use output of the encoder without projection
    - for multimodal finetuning: use output of the shared layer without projection
    - for retrieval: use output of the projection layer
- why even use shared/fuzed layers, why not directly use the same approach as in CLIP? -> test the following: just train a text model with beit-2

-> (CLIP like) just train a text model to regress cls token output of final beit-2 layer -> use blocks of d2v2 text to init text model (generally one could take any pretrained text model) -> freeze embedding layer + pos embedding layer -> has the advantage that max possible context stays 512 tokens, so no need to interpolate for downstream tasks, even if we now use just 64 tokens

- have seperate projection layer for cls token of text encoder and image encoder
- contrastive loss is done based on the cosine similarity of the projected cls tokens
- output of the cls token of the shared multimodal layer(s) is ignored for now
- also means that we use the projected cls tokens for retrieval, which is now not zero-shot anymore, as we explicitly train the model to maximize the cosine similarity between the cls tokens of an matching image-text pair
    - in that regard what FLAVA claims is not true, as they name their results on cross-modal retrieval for COCO and Flickr30K zero-shot, but pretrain their model using a contrastive loss
- we have 5 encoder layers for each modality, and two shared layers

-> means we now use 5 layers + projection for contrastive learning, and therefore for retrieval -> performance result will be interesting

- the question is in which case we will then utilize the output of the shared layers (cls tokens)
- for multimodal task we would use the output of the projections of the encoder's cls tokens
- for unimodal tasks we would use the output of the encoders without the projections
- therefore better option would be to use the cls token of the last layer output, which is a shared one, and project this into the shared space
- even though the representation should already be shared at this point -> so no projection necessary
- for modality specific tasks -> use output of the corresponding encoder
- for multimodal tasks -> use output of the shared layer

following options:

- no shared layers, seperate image and text encoders and two linear projections to shared space (CLIP)
- shared layers, seperate image and text encoders and one linear projection to contrast space
  - use output of shared layer for multimodal tasks, output of encoder for modality specific tasks
  - output of projection for retrieval

CLIP: no shared layers, seperate image and text encoders and two linear projections to shared space FLAVA: shared layers, seperate image and text encoders, two linear projections to shared space for image and text encoder, two lin projections for image/text to multimodal space for mm encoder, on unimodal downstream classification tasks: use output of respective encoder withouth projection

- we do not do it exactly the same as in FLAVA, as our shared layers are not (yet) for referencing multimodal tasks, but for aligning the modalities on the "concept" level, so use single projection layer for contrastive learning on this output
- also test exactly as in flava
  - have one projection layer for the final output of the cls token, stemming from the shared layer(s)

2.3.2.1 Contrastive Learning with Memory Bank

2.3.2.2 Decaying Memory Bank

2.3.3 Importance of the Teacher Model

- BEiT-2 vs. D2V2 Image shows significant difference in performance
- Model distilled from BEiT-2 teacher outperforms the one from D2V2 Image teacher by a large margin
- teacher model size is around the same -> both use ViT-B/16, BEiT-2 around one percent better on Imagenet-1k after finetuning
- too small of a difference that this could be the reason for the large difference in performance
- most likely the handling of the CLS token, which is regressed by our students, is the reason
  - D2V2 Image introduces special CLS loss to aggregate as much (global) information as possible
    - cls token regresses mean activations of all patches
    - was inspired by BEiT-2
  - BEiT-2 introduces a bottleneck to force the model to push as much information as possible towards the cls token
  - latter seems to be more effective
- which teacher to use does make a difference!

A Supplementary Material Images

– Supplementary Material –

B Supplementary Material Source Code

Bibliography

[Lin+14] T.-Y. Lin *et al.*, "Microsoft COCO: Common Objects in Context," in *Computer Vision - ECCV 2014 - 13th European Conference, Zurich, Switzerland, September 6-12, 2014, Proceedings, Part V*, D. J. Fleet, T. Pajdla, B. Schiele, and T. Tuytelaars, Eds., in Lecture Notes in Computer Science, vol. 8693. Springer, 2014, pp. 740–755.

[Wan+23] W. Wang *et al.*, "Image as a Foreign Language: BEIT Pretraining for Vision and Vision-Language Tasks," in *2023 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2023, pp. 19175–19186. doi: 10.1109/CVPR52729.2023.01838.

[AVT17] Y. Aytar, C. Vondrick, and A. Torralba, "See, Hear, and Read: Deep Aligned Representations," *arXiv preprint arXiv:1706.00932*, 2017, [Online]. Available: https://arxiv.org/abs/1706.00932

[OKB11] V. Ordonez, G. Kulkarni, and T. L. Berg, "Im2text: Describing images using 1 million captioned photographs," in *Advances in Neural Information Processing Systems 24: 25th Annual Conference on Neural Information Processing Systems 2011. Proceedings of a meeting held 12-14 December 2011, Granada, Spain*, J. Shawe-Taylor, R. S. Zemel, P. L. Bartlett, F. C. N. Pereira, and K. Q. Weinberger, Eds., 2011, pp. 1143–1151.

[Sha+18] 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 the 56th Annual Meeting of the Association for Computational Linguistics, ACL 2018, Melbourne, Australia, July 15-20, 2018, Volume 1: Long Papers*, I. Gurevych and Y. Miyao, Eds., 2018, pp. 2556–2565.

[Kri+17] R. Krishna *et al.*, "Visual Genome: Connecting Language and Vision Using Crowdsourced Dense Image Annotations," *Int. J. Comput. Vision*, vol. 123, no. 1, pp. 32–73, May 2017.

[Soc+13] R. Socher *et al.*, "Recursive Deep Models for Semantic Compositionality Over a Sentiment Treebank," in *Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing*, Seattle, Washington, USA: Association for Com-

putational Linguistics, Oct. 2013, pp. 1631–1642. [Online]. Available: https://www.aclweb.org/anthology/D13-1170

[WSB18] A. Warstadt, A. Singh, and S. R. Bowman, "Neural Network Acceptability Judgments," *arXiv preprint arXiv:1805.12471*, 2018.

[May21] P. May, "Machine translated multilingual STS benchmark dataset.," 2021. [Online]. Available: https://github.com/PhilipMay/stsb-multi-mt

[DB05] W. B. Dolan and C. Brockett, "Automatically constructing a corpus of sentential paraphrases," in *Proceedings of the International Workshop on Paraphrasing*, 2005.

[Raj+16] P. Rajpurkar, J. Zhang, K. Lopyrev, and P. Liang, "SQuAD: 100,000+ Questions for Machine Comprehension of Text," in *Proceedings of EMNLP*, Austin, Texas: Association for Computational Linguistics, 2016, pp. 2383–2392.

[Wan+19] A. Wang, A. Singh, J. Michael, F. Hill, O. Levy, and S. R. Bowman, "GLUE: A Multi-Task Benchmark and Analysis Platform for Natural Language Understanding," 2019.

[Ben+09] L. Bentivogli, I. Dagan, H. T. Dang, D. Giampiccolo, and B. Magnini, "The Fifth PASCAL Recognizing Textual Entailment Challenge," 2009.

[DGM06] I. Dagan, O. Glickman, and B. Magnini, "The PASCAL recognising textual entailment challenge," *Machine learning challenges. evaluating predictive uncertainty, visual object classification, and recognising tectual entailment*. Springer, pp. 177–190, 2006.

[Gia+07] D. Giampiccolo, B. Magnini, I. Dagan, and B. Dolan, "The third PASCAL recognizing textual entailment challenge," in *Proceedings of the ACL-PASCAL workshop on textual entailment and paraphrasing*, 2007, pp. 1–9.

[Bar+06] R. Bar Haim *et al.*, "The second PASCAL recognising textual entailment challenge," 2006.

[WNB18] A. Williams, N. Nangia, and S. R. Bowman, "A Broad-Coverage Challenge Corpus for Sentence Understanding through Inference," in *Proceedings of NAACL-HLT*, 2018.

[You+14] P. Young, A. Lai, M. Hodosh, and J. Hockenmaier, "From image descriptions to visual denotations: New similarity metrics for semantic inference over event descriptions," *Transactions of the Association for Computational Linguistics*, vol. 2, pp. 67–78, Feb. 2014.

[Sin+21] A. Singh *et al.*, "FLAVA: A foundational language and vision alignment model," *CoRR*, 2021, [Online]. Available: https://arxiv.org/abs/2112.04482

[Rad+21] A. Radford *et al.*, "Learning transferable visual models from natural language supervision," in *Proceedings of the 38th International Conference on Machine Learn-*

ing, ICML 2021, 18-24 July 2021, Virtual Event, M. Meila and T. Zhang, Eds., in Proceedings of Machine Learning Research, vol. 139. PMLR, 2021, pp. 8748–8763.

[Bao+22] H. Bao *et al.*, "VLMo: Unified Vision-Language Pre-Training with Mixture-of-Modality-Experts," in *Advances in Neural Information Processing Systems*, 2022. [Online]. Available: https://openreview.net/forum?id=bydKs84JEyw

[Vas+17] A. Vaswani *et al.*, "Attention is all you need," in *Proceedings of the 31st International Conference on Neural Information Processing Systems*, in NIPS'17. Long Beach, California, USA: Curran Associates Inc., 2017, pp. 6000–6010.

[Li+17] A. Li, A. Jabri, A. Joulin, and L. van der Maaten, "Learning visual n-grams from web data," in *Proceedings of the IEEE International Conference on Computer Vision*, 2017, pp. 4183–4192.

[Wu+18] Z. Wu, Y. Xiong, S. X. Yu, and D. Lin, "Unsupervised Feature Learning via Non-parametric Instance Discrimination," in *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2018, pp. 3733–3742. doi: 10.1109/CVPR.2018.00393.