where $\mathcal{L}_{ling}$ is the Skip-gram loss function and $\mathcal{L}_{vision}$ is the additional visual loss for the target word w_t. In particular, $\mathcal{L}_{vision}$ has the form of a hinge loss, the goal of which is to make the (vectorial) linguistic representation of a certain word more similar to its visual representation:

$$\mathcal{L}_{vision}(w_t) = - \sum_{w' \sim P_n(w)} (max(0, \gamma - cos(z_{w_t}, v_{w_t}) + cos(z_{w_t}, v_{w'}))$$

where $v_{w'}$ is a visual representation of a randomly chosen word $w^{'}$ (drawn from a probability distribution $P_n(w)$) used as negative sample, v_{w_t} is the corresponding visual vector and z_{w_t} is the target multimodal word representation which has to be learned by the model. It is nothing more than a linear transformation of a word representation u_{w_t}: $z_{w_t} = M^{u \to v} u_{w_t}$ and $M^{u \to v}$ is a cross-modal mapping matrix from linguistic inputs to a visual representation. It is important to remark that during training, for words which do not have associated images, $\mathcal{L}_{vision}$ gets set to zero. When this cross-modal mapping matrix is estimated, it is then possible to find a visual representation for new words, which do not have a related image in the training set: the model allows to *imagine* new words. This is what is meant with grounded space: a perceptual (visual, in this case) space where a word is *grounded*, put in context.

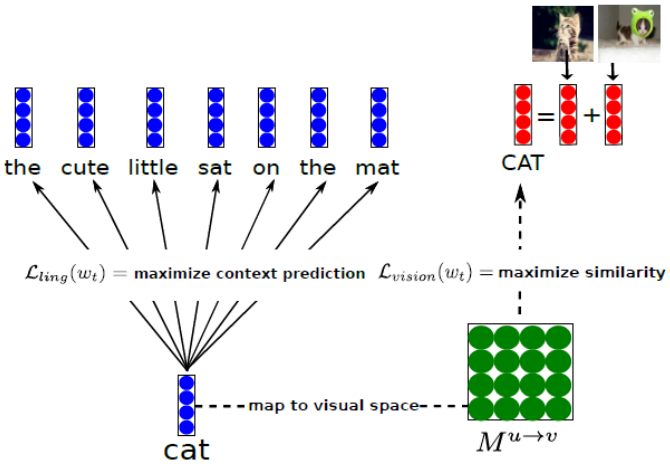


FIGURE 3.40: From Lazaridou et al. (2015). The linguistic embedding of the word 'cat' is mapped to a visual space, such that the similarity of vector representations of words and associated images is maximized.

Similar instances of a cross-modal mapping can be found for example in Kottur et al. (2016) (a multimodal extension of the CBOW model specification of word2vec) and in Collell et al. (2017), where visual features are obtained from the forward pass of a CNN, pre-trained on ImageNet (Deng et al. (2009)) and

a mapping function from the textual space to the visual space is obtained as a result of the training process. Also in this case it is possible to generate a visual representation from the embedding of a certain word, not necessarily present in the training set. In particular, they propose two specifications of the mapping function: a simple linear mapping and neural network with a single hidden layer. Last but not least, Hill and Korhonen (2014) recognize that concrete nouns are more likely to have a visual representation. For this reason, they map a set of concrete words (CSLB, Devereux et al. (2014)) to "bags of perceptual/visual features" and every time one of these words is encountered during training, the Skip-gram model they are using stops training on that sentence and instead continues the training on a newly created "pseudo-sentence", which takes into consideration the aforementioned bag of perceptual features. This list is unfortunately not exhaustive and there are other models with similar ideas, for example Ailem et al. (2018) or Kiros et al. (2018).

The aforementioned papers and related models focus on the modeling of semantics of words. Nonetheless, there are models designed to address tasks at sentence-level, such as sentiment analysis or sentence entailment. Kiela et al. (2017) employ a bidirectional Long Short-Term Memory (LSTM, Hochreiter and Schmidhuber (1997)) architecture to model sentence representations, in order to gain information from the text in both directions. The goal is again to encode a sentence and ground it in an image. Textual embeddings are obtained with GloVe (Pennington et al. (2014)) and they are then projected on a grounded space with a linear mapping. This grounded word vector serves as input for the bidirectional LSTM, which is trained together with the linear mapping. Their model is versatile and depending on the loss function specification, it can not only propose alternative captions to an image (which is a way to frame sentence equivalence tasks) but also predict captions from images or perform both tasks at the same time. This last point highlights an important characteristic of many of the models discussed in this subchapter: even though the focus is on the empowerment of pure language models with the addition of visual elements, some of the models discussed here can be used for purposes other than pure language tasks. The control over which task is performed is usually exercised by either specifying different loss functions (as in the last model described) or setting properly certain hyperparameters (such as in the previously described model by Silberer and Lapata (2014)).

3.3.5 The Transformers Era

A turning point for the field of NLP was Vaswani et al. (2017b)'s paper "Attention is all you need", where the authors proposed for two machine translation tasks a novel architecture, the Transformer (not to be confused with the giant robots from the Michael Bay's blockbuster movies!), which leverages only the attention mechanism. Even though an exhaustive description of the

Transformer architecture is beyond the scope of this subchapter, it is worth mentioning why they became so popular over the past four years in the field of NLP (among others), in comparison to Recurrent Neural Networks (RNNs) and Long Short-Term Memory networks (LSTMs).

Well, the three main properties of Transformers are the following:

- Self-Attention
- Parallel input processing
- Positional embeddings[2]

When feeding for example a textual sentence to a RNN, the network deals with one word after the other in a sequential fashion and one of the known issues is the fact that information contained in earlier parts of the sequence tend to "fade away" as the sentence is analyzed further: newer inputs carry a larger influence on the outputs at a given step. LSTMs try to mitigate this problem by introducing a component called "gate", which regulates the information flow, namely which information from the past inputs need to be "remembered" by the model. The goal is to capture long-term dependencies among different parts of the sentence fed into the model.
On the contrary, thanks to the Self-Attention mechanism, at each step Transformers can access previous steps, thus limiting to a minimum the loss of information. Moreover, inputs are processed not sequentially but all at the same time, thus allowing to capture dependencies by looking at the sentence *as a whole* and this could make a fundamental difference in many downstream applications: for example in the German language, in dependent clauses ("Nebensaetze"), the verb comes at the end of the phrase but it determines the verbal case of the nouns that come *before* the verb. Thus Transformer could potentially capture the dependencies between the verb coming at the end of the sentence and the words at the beginning. Lastly, Transformers encode for every input information on its position within a sentence, since it is often the case, that the importance and meaning of a certain word varies depending on its position within a sentence. These were the Transformers, in a nutshell.

But Transformers did not only bring a change of paradigm in terms of architectures. First, while for models in the pre-Transformers era described before, the focus was on the ability of word embeddings to capture similarity among words, now the focus has shifted more on downstream tasks (more on this later in the evaluation section), encompassing not only pure linguistic ones but also tasks with visual components, such as for example, visual question answering. It is now more difficult (but not impossible) to draw a line between models where "images support pure language models" (the object of this subchapter) and models which could be actually categorized as "vision and language" models but they can be employed also to solve pure linguistic tasks. This issue brings

[2]It may be argued that this point is a necessity to be able to work on sequences rather than a strength.

another peculiarity of many Transformers-base models, namely their "universal vocation": without loss of generality we could say that the idea is now to design powerful (multimodal) pre-training (mostly *self-supervised*) tasks capable of generating task-agnostic representations, whose encoded knowledge can be efficaciously transferred to diverse downstream tasks, limiting the amount of labeled data necessary to fine-tune the models (this is the so-called *few-shot learning*).

Let's briefly discuss two examples, Flava (Singh et al. (2022)) and UniT (Hu and Singh (2021a)). Flava has two separate encoders for images and text and a multimodal encoder, all based on the Vision Transformer (Dosovitskiy et al. (2020a)). Unimodal pre-training consists of masked image modeling (where a set of image patches are to be reconstructed from other unmasked image patches) and masked language modeling. Multimodal pre-training tasks consist instead of a global contrastive loss (maximization of cosine similarities between paired images and text), a masked multimodal modeling (where image patches and text tokens are masked) and an image-text matching task. The model is pre-trained jointly on unimodal and multimodal datasets and then evaluated (fine-tuned) on 22 vision tasks, 8 pure linguistic tasks and 5 vision and language tasks.
UniT has an image encoder and a text encoder, a multimodal domain-agnostic decoder and task-specific heads. There is no pre-training on multimodal data and the model is trained end-to-end on 7 tasks (vision, language and vision an language) and 8 datasets, with the idea that solving different tasks across domains in a jointly fashion should prevent general knowledge from being lost due to fine-tuning over particular downstream tasks.

These two examples clearly show what it is meant by "universal vocation" of many modern Transformer-based models. But there are still models specifically designed to solve pure language tasks and in the following pages, two of them will be described.

3.3.5.1 Vokenization

It is often difficult for a child to describe the meaning of a certain word. A child might not be able to describe what a lion is but if he is given pictures of different animals he might be very well able to point at the picture of a lion. *Visual pointing* could thus act as a form of supervision to natural language. Is it possible to build within a pure language model a form of visual supervision, which mimics the visual pointing often adopted by children? This is exactly the problem that Tan and Bansal (2020) try to address: how to associate to each textual representation (token) a visual representation (Voken).

Let's suppose we had a dataset of word(token)-image pairs. We could integrate in the pre-training framework of pure language models the following *Voken-Classification* task:

$$\mathcal{L}_{VOKEN-CLS}(s) = -\sum_{i=1}^{l} log\ p_i(v(w_i;s)|s)$$
$$\mathbf{h}_1, \mathbf{h}_2, ..., \mathbf{h}_l = languagemodel(w_1, w_2, ..., w_l)$$
$$p_i(v|s) = softmax_v\{W\mathbf{h}_i + b\}$$

where $\{h_i\}$ is the feature representation of each token in a sentence $s = \{w_i\}$ extracted from a language model (such as BERT) and the vokens originate from a **finite** set of images X. Each h_i is then transformed into a probability distribution through a softmax layer, with the voken-classification loss defined as the negative log-likelihood of all related vokens.
The model architecture would then be:

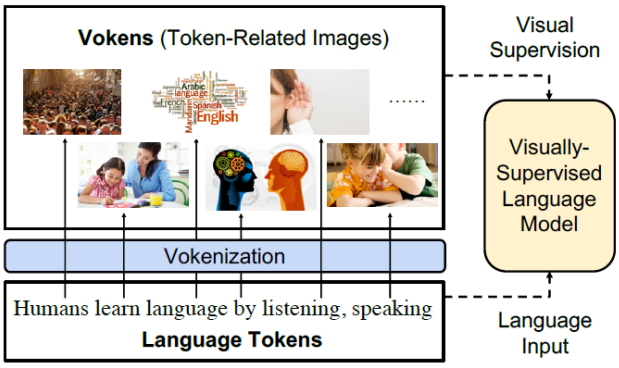


FIGURE 3.41: From Tan and Bansal (2020). Visually supervised the language model with token-related images, called Vokens.

Everything sounds fantastic! There is only one small pitfall: a set of X of images for all tokens does not exist! Could we find a proxy for such a set? One might consider image-captioning datasets such as MS COCO (Lin et al. (2014b)). But also this suboptimal solution is problematic.

The *Grounding Ratio* is defined as the proportion of tokens in a dataset which are related to a specific visual representation (i.e. the tokens are *visually grounded*), such as "dog", "table" and the like. In figure 3.42 it is striking that only around one third of tokens contained in pure language corpora such Wiki103, English Wikipedia and CNN/DM are visually grounded in image captioning datasets[3]. It is not possible to rely (only) on image captioning datasets to build the Voken-Classification task. But the fact that a word/token does not have a visual representation in one of these datasets, it does not mean that it is not possible to visually represent the word/token. Would it be possible to associate images to words/tokens not directly visually grounded? Well, the answer is yes!

[3]From an operative point of view, the authors consider a token type "visually grounded" if it has more than 100 occurrences in MS COCO

Dataset	# of Tokens	# of Sents	Vocab. Size	Tokens #/ Sent.	1-Gram JSD	2-Gram JSD	Grounding Ratio
MS COCO	7.0M	0.6M	9K	11.8	0.15	0.27	54.8%
VG	29.2M	5.3M	13K	5.5	0.16	0.28	57.6%
CC	29.9M	2.8M	17K	10.7	0.09	0.20	41.7%
Wiki103	111M	4.2M	29K	26.5	0.01	0.05	26.6%
Eng Wiki	2889M	120M	29K	24.1	0.00	0.00	27.7%
CNN/DM	294M	10.9M	28K	26.9	0.04	0.10	28.3%

FIGURE 3.42: From Tan and Bansal (2020). Statistics of image-captioning dataset and other natural language corpora. VG, CC, Eng Wiki, and CNN/DM denote Visual Genome, Conceptual Captions, English Wikipedia, and CNN/Daily Mail, respectively. JSD represents Jensen–Shannon divergence to the English Wikipedia corpus.

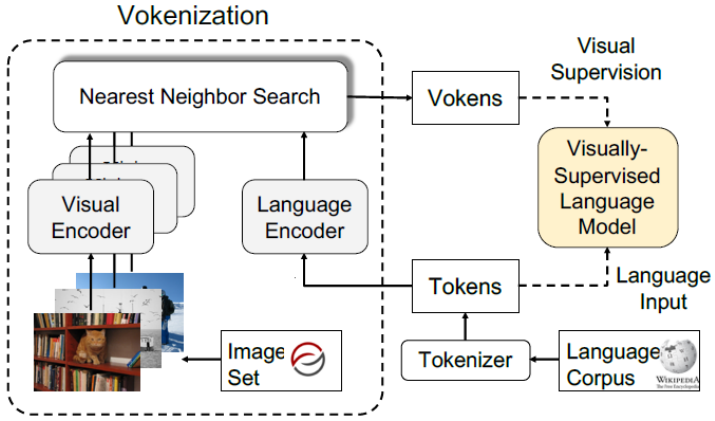


FIGURE 3.43: From Tan and Bansal (2020). The Vokenization process. A contextualized image (visual token, Voken) is retrieved for every token in a sentence and with this visual token, visual supervision is performed.

The **Vokenization** is a process to *assign* every token w_i contained in a sentence s to a visual representation (called *voken*) originating not from a generative model but rather from a finite set of images $X = \{x_1, ..., x_n\}$. The voken $v(w_i; s)$ is the image from X which maximizes the following *Relevance Score Function*:

$$v(w_i; s) = arg\ max_{x \in X} r_{\theta^*}(w_i, x, s)$$

This function takes into account not only the token w_i itself, but also the context (the sentence) and it is parametrized by θ with θ^* being the optimal value (which has to be estimated).

3.3.5.1.1 The Relevance Score Function: Model, Training, Inference

The Relevance Score Function is defined as the inner product of the language feature representation $f_\theta(w_i, s)$ and the visual feature representation $g_\theta(x)$:

$$f_\theta(w_i, s)^T g_\theta(x)$$

Supposing $h_1, ..., h_l$ and e are the embeddings originating from pre-trained language and visual encoders respectively (in the paper the authors use BERT and ResNeXt), the language and visual representations are obtained first by applying multi-layer perceptrons w_mlp_θ and x_mlp_θ to downproject the embeddings from the pre-trained models to a common vector space and secondly they are normalized (with L2-Norm):

$$\mathbf{f}_\theta(w_i; s) = \frac{w\_mlp_\theta(\mathbf{h}_i)}{||w\_mlp_\theta(\mathbf{h}_i)||}$$
$$\mathbf{g}_\theta(x) = \frac{x\_mlp_\theta(\mathbf{e})}{||x\_mlp_\theta(\mathbf{e})||}$$

With respect to the training of the model, to estimate the optimal value for the parameter θ, image-captioning datasets, which are collections of sentence-image pairs, are employed. Operationally, for every sentence s_k associated to image x_k in the image-captioning dataset, each token w_i in s is associated to x_k and the *hinge loss* is used to estimate the optimal value of θ^*:

$$\mathcal{L}_\theta(s, x, x') = \sum_{i=1}^{l} max(0, M - r_\theta(w_i, x, s) + r_\theta(w_i, x', s))$$

The goal is to maximize the Relevance Score Function between aligned token-image pairs $(w_i, x; s)$ and to minimize the score for unaligned pairs $(w_i, x^{'}; s)$ by at least a margin M, with $x^{'}$ being a randomly sampled image from the image captioning dataset **not** associated to sentence s.

Once we have the language feature representation $f_\theta(w_i, s)$ for each token in our language corpus and the optimal estimate of θ, how is it possible to find the image x encoded with the visual feature representation $g_\theta(x)$, which maximizes the Relevance Score Function? As said earlier, the function is expressed as the inner product of the textual and visual representations and since the feature vectors have euclidean norm equal to 1, the inner product maximization problem is equivalent to a nearest neighbor search problem. It is just sufficient to find the vector $g_\theta(x)$ which is the nearest neighbor of $f_\theta(w_i, s)$[4].

With this process, it is thus possible to assign a visual representation, a voken, to any word/token in a language corpus, pooling from a finite set of images. The problem of the low Grounding Ratio outlined above is solved and the Voken-Classification task could be integrated in the pre-training framework

[4]The proof is straightforward. Let $X \in \mathbb{R}^l$ and have euclidean norm equal to 1, which means $||X||_2 = 1$. In the nearest neighbor search we need to find the vector $Y \in \mathbb{R}^l$, also with norm equal to 1, which has minimal euclidean distance with X. This is the quantity to

of any pure language model. Moreover, the authors propose a method called *Revokenization*, which allows to transfer vokens generated using a particular tokenizer to frameworks which employ other tokenizers.

3.3.5.2 One Step Further: The Power Of Imagination

Wikipedia defines *imagination* as "the production or simulation of novel objects, sensations, and ideas in the mind without any immediate input of the senses". Indeed, humans do not only associate words with real images, but also leverage the ability to *imagine* words/concepts: imagination can help the human brain solve problems with limited supervision or sample points by empowering its generalization capabilities. Until now we discussed language models supported by visual information in form of *real* images (e.g. those retrieved from image-captioning datasets). But with the recent advancements in the field of generative models for images, it is for sure worth investigating if these generative models can help pure language models to produce better representations of words. In particular, the framework proposed by Lu et al. (2022), **iACE (Imagination-Augmented Cross-Modal Encoder)** will now be discussed: the idea is simply to use a generative model to obtain a visual representation of a textual input and then use these imagined representations as "imagination supervision" to pure language models.

This framework has two main components:

- the **imagination generator** G: given an input text x, VQGAN (Esser et al. (2021)) is used to render an "imagination" i of x and CLIP (Radford et al. (2021a)) is used to see how well the generated image i is aligned to the input text x. This generative framework is known as VQGAN+CLIP
- **Cross-modal Encoder** E_c: the input text and the rendered imagination are firstly encoded with a language and a visual encoder respectively and then

be minimized:

$$
\begin{aligned}
d(X,Y) &= \sqrt{\sum_{i=1}^{l}(x_i - y_i)^2} \\
&\overset{squared}{=} \sum_{i=1}^{l} x_i^2 + \sum_{i=1}^{l} y_i^2 - 2\sum_{i=1}^{l} x_i y_i \\
&= ||X||_2^2 + ||Y||_2^2 - 2X^T Y \\
&\overset{Norm-1}{=} 1 + 1 - 2X^T Y \\
&= 2(1 - X^T Y)
\end{aligned}
$$

And through these simple algebraic manipulations, it is possible to see that minimizing the euclidean distance between X and Y is equivalent to maximize $X^T Y$, which is the inner product. This proves the equivalence between inner product maximization and nearest neighbor search.

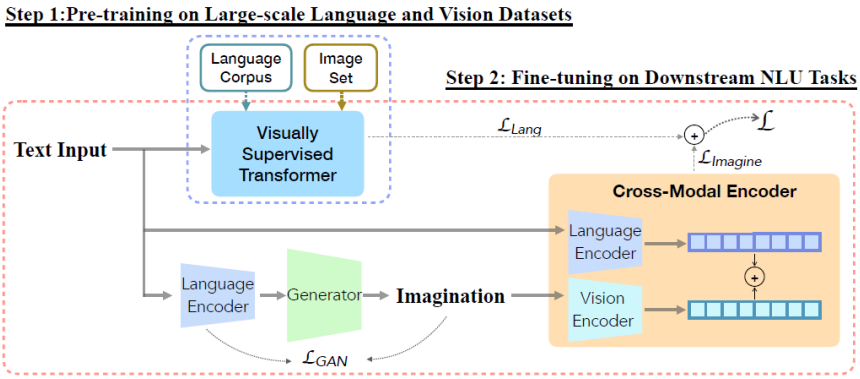


FIGURE 3.44: From Lu et al. (2022). The generator G visualize imaginations close to the encoded texts by minimizing $\mathcal{L}_{GAN}$. The cross-modal encoder E_c learns imagination-augmented language representation. Two-step learning procedure consists of: 1) pre-train a Transformer with visual supervision from large-scale language corpus and image set, 2) fine-tune the visually supervised pre-trained Transformer and the imagination-augmented cross-modal encoder on downstream tasks.

CLIP is employed as cross-modal encoder with inputs being text-imagination pairs

The learning procedure is composed of two main steps (depicted in figure 3.44): the first step consists in the pre-training of a visually supervised Transformer. In particular, the Voken-Classification task described before is employed, alongside a masked language modeling task. This is the baseline model, where no information from the "imagination" procedure comes yet into play. The second step is the *imagination-augmented fine-tuning* with two downstream datasets D (GLUE, Wang et al. (2018) and SWAG, Zellers et al. (2018)).
On one side, the visually-supervised Transformer (the baseline) relies only on the textual input during the fine-tuning phase and the following loss function is employed:

$$\mathcal{L}_{Lang} = -\sum_{j=1}^{|D|}\sum_{k=1}^{K} y_k \; log \; p_k(d_j(t)|D)$$

On the other hand, the *iACE* is trained to minimize the following cross-entropy loss:

$$\mathcal{L}_{Imagine} = -\sum_{j=1}^{|D|}\sum_{k=1}^{K} y_k \; log \; p_k(d_j(t,v)|D)$$

with t and v being the textual and imagined features representations respectively, j indicates the j-th data sample in dataset belonging to dataset D, K is the number of classes and p_k is the conditional distribution of d_j. Training takes place in a jointly fashion and both losses, the imagination-augmented one $\mathcal{L}_{Imagine}$ and the pure language loss $\mathcal{L}_{Lang}$ are linearly combined, with λ being a balance factor:

$$\mathcal{L} = \lambda\mathcal{L}_{Imagine} + (1 - \lambda)\mathcal{L}_{Lang}$$

To sum up, this model-agnostic framework uses *generated images* for visual supervision and could be integrated on top of pure language models (such as BERT) or visually supervised models (such as the Voken model, which uses Vokens, real images for visual supervision).

3.3.6 Was It Worth?

In this subchapter we investigated how visual inputs can support pure language models in capturing the semantics of words. We started with simple concatenation of linguistic and visual features and ended up with Transformer-based models, which are able to shape different word embeddings for the same word by taking into account also the context (the sentence). But now the question arises: with the addition of visual information, do we obtain word embeddings that are better than those from pure language models? In other words, is what we all have so far discussed worth? Well, as it is often the case in scientific research, the answer is: "it depends!"

Individual evaluation of each single model might not be ideal because each model has its peculiarities and it is impractical to make a direct comparison among them. It is more useful to capture and discuss the themes which are common to many models, in order to understand their strengths and weaknesses. This is how we will proceed and we will also differentiate between evaluation before Transformers and evaluation after Transformers.

3.3.6.1 Evaluation In The Pre-Transformers Era

Before the advent of Transformers, the evaluation focus was on the degree of alignment between learned semantic representations (word embeddings) and representations by human speakers, in form of correlation between model-based and human-based word-similarity judgments. Three main types of similarity are usually considered:

- Semantic similarity, e.g. "pasta is similar to rice"
- Semantic relatedness, e.g. "Bear is related to mountain"
- Visual similarity, e.g. "cucumbers look like zucchinis"

The evaluation pipeline could be summarized as follows:

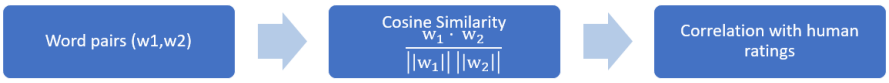


FIGURE 3.45: Pipeline for intrisinsic evaluation of semantic representations. In the first step, the cosine similarity between two word embeddings w1 and w2 is used as similariry measure and in a second step, the correlation with human speakers'assessment is computed to gauge the quality of the embeddings. The higher the correlation, the better the embeddings.

Word embeddings are vectors and to measure the degree of similarity between two vectors, the *Cosine Similarity* is often used in the literature. In an ideal setting, we would have word embeddings with the following characteristics: if two words are semantically similar, the two embedding vectors should be similar and their cosine similarity should go towards 1. If the two words are unrelated, the embedding vectors should be orthogonal to each other and as a consequence, the cosine similarity should go towards zero. Lastly, if two words are negatively related, the two embedding vectors should point at opposite directions and the cosine similarity should go towards -1. Once these similarity measures between word pairs are computed, in order to measure the quality of the embeddings several benchmarks can be employed, such as MEN (Bruni et al. (2014)), WordSim353 (Agirre et al. (2009)) and SimLex999 (Hill et al. (2015)). These datasets could be described as collections of word pairs and associated similarity ratings by human speakers. Operationally, this means that real people were asked if a pair of words was related or not and to which degree, on a scale between -1 (negatively related) to +1 (semantically equivalent). The higher the correlation between the cosine similarity and the similarity judgments by humans, the higher the quality of the word embeddings. Having done this methodological premise, let's discuss the performance of these pre-Transformer models!

Since the goal of these models is to enhance pure language models with the addition of visual inputs, the baseline in the evaluation is always one (or more) pure language model(s). Well, do visually grounded embeddings outperform non-grounded ones? What emerges from virtually all papers is that visual grounding can actually help get a better semantic representation of *concrete* concepts, such as "cat", "table", "bicycle", whereas they do not help much with the representation of abstract concepts such as "love" and "peace".

3.3.6.2 Evaluation In The Post-Transformers Era

A limitation of the *intrinsic* evaluation metrics is the high degree of subjectivity: the *similarity* between two concepts depends in many instances on the experience, cultural background and preferences of the human observers. This is why the evaluation focus has now shifted to a more *extrinsic* dimension: how well do the models perform in downstream tasks? The problem of the "lack

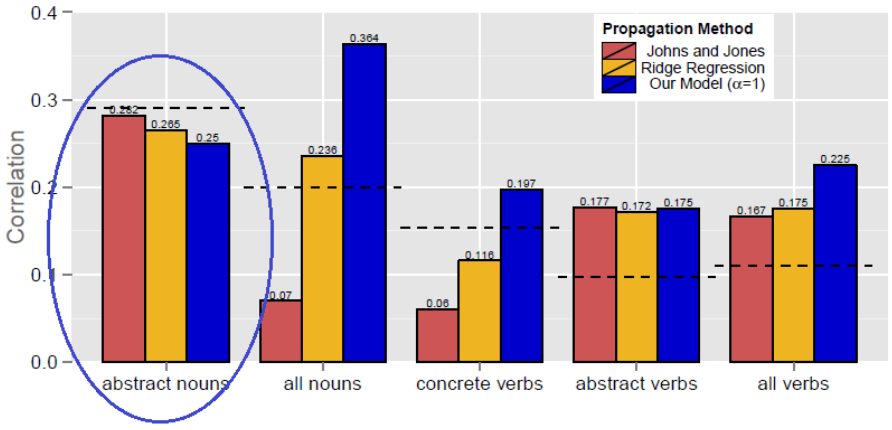


FIGURE 3.46: From Hill and Korhonen (2014): Each bar represents a different model settings and the dashed line indicates the pure linguistic benchmark model. In figure 3.46 we can see that pure language models still perform better than models with visual inputs when it comes to the representation of abstract *nouns*. Another example is Kiela et al. (2017): they found that their models perform better when tested on datasets with a higher degree of concreteness and the same conclusion is reached by Collell et al. (2017), which state that visual information can empower the representations of concepts that are to a certain extent visual. To sum up, effective semantic representation of abstract concepts constitute the main limitation common to many of the models discussed in this section.

of objectivity" is thus solved because on downstream tasks there is no room for opinions. The datasets used to train the models are also different and the most widely used are:

- GLUE (Wang et al. (2018)): 9 tasks, including single-sentence tasks (e.g. sentiment analysis), similarity tasks (e.g. paraphrasing), inference tasks (e.g. textual entailment)
- SQuAD (Rajpurkar et al. (2016)): question/answer pairs
- SWAG (Zellers et al. (2018)): multiple choice questions about grounded situations

As previously discussed, many Transformer-based models have universal vocation: they are built to solve a heterogeneous range of tasks from the language and vision domain. If we thus consider only performance on pure language tasks, the following two tables from Tan and Bansal (2020) are insightful:

It is straightforward: unlike in the pre-Transformers Era, where grounded word embeddings could improve performance over baselines, Transformer-based universal models **do not** outperform pure language models such as BERT or RoBERTa. Nonetheless, the addition of visual supervision (the Voken-

Model	Init. with BERT?	Diff. to BERT Weight	SST-2	QNLI	QQP	MNLI
ViLBERT (Lu et al., 2019)	Yes	0.0e-3	90.3	89.6	88.4	82.4
VL-BERT (Su et al., 2020)	Yes	6.4e-3	90.1	89.5	88.6	82.9
VisualBERT (Li et al., 2019)	Yes	6.5e-3	90.3	88.9	88.4	82.4
Oscar (Li et al., 2020a)	Yes	41.6e-3	87.3	50.5	86.6	77.3
LXMERT (Tan and Bansal, 2019)	No	42.0e-3	82.4	50.5	79.8	31.8
$\text{BERT}_{\text{BASE}}$ (Devlin et al., 2019)	-	0.0e-3	90.3	89.6	88.4	82.4
$\text{BERT}_{\text{BASE}}$ + Weight Noise	-	6.5e-3	89.9	89.9	88.4	82.3

FIGURE 3.47: From Tan and Bansal (2020). Statistics of image-captioning dataset and other natural language corpora. VG, CC, Eng Wiki, and CNN/DM denote Visual Genome, Conceptual Captions, English Wikipedia, and CNN/Daily Mail, respectively. JSD represents Jensen–Shannon divergence to the English Wikipedia corpus.

Method	SST-2	QNLI	QQP	MNLI	SQuAD v1.1	SQuAD v2.0	SWAG	Avg.
$\text{BERT}_{\text{6L/512H}}$	88.0	85.2	87.1	77.9	71.3/80.2	57.2/60.8	56.2	75.6
$\text{BERT}_{\text{6L/512H}}$ + Voken-cls	89.7	85.0	87.3	78.6	71.5/80.2	61.3/64.6	58.2	76.8
$\text{BERT}_{\text{12L/768H}}$	89.3	87.9	83.2	79.4	77.0/85.3	67.7/71.1	65.7	79.4
$\text{BERT}_{\text{12L/768H}}$ + Voken-cls	**92.2**	**88.6**	**88.6**	**82.6**	**78.8/86.7**	68.1/71.2	**70.6**	**82.1**
$\text{RoBERTa}_{\text{6L/512H}}$	87.8	82.4	85.2	73.1	50.9/61.9	49.6/52.7	55.1	70.2
$\text{RoBERTa}_{\text{6L/512H}}$ + Voken-cls	87.8	85.1	85.3	76.5	55.0/66.4	50.9/54.1	60.0	72.6
$\text{RoBERTa}_{\text{12L/768H}}$	89.2	87.5	86.2	79.0	70.2/79.9	59.2/63.1	65.2	77.6
$\text{RoBERTa}_{\text{12L/768H}}$ + Voken-cls	**90.5**	**89.2**	**87.8**	**81.0**	**73.0/82.5**	**65.9/69.3**	**70.4**	**80.6**

FIGURE 3.48: From Tan and Bansal (2020). Fine-tuning results of different pre-trained models w/ or w/o the voken classification task (denoted as"Voken-cls").

Classification task) in the pre-training framework can boost performance above the level of pure language models.

Pezzelle et al. (2021) analyzed the *intrinsic* quality of embeddings of some vision and language ("universal") models:

From this *intrinsic* evaluation perspective (which was popular in the pre-Transformers Era), vision and language models do not generally outperform domain-specific models such as BERT and also in this case the only real competitor of pure language models is a model with visual supervision (again, Vokenization).

The bar plots depict correlation between human- and model-based similarity ratings, differentiating between the most *concrete* concepts contained in a

model	*input*	*Spearman ρ correlation (layer)*				
		RG65	WS353	SL999	MEN	SVERB
BERT-1M-Wiki*	L	0.7242 (1)	0.7048 (1)	0.5134 (3)	–	0.3948 (4)
BERT-Wiki *ours*	L	0.8107 (1)	0.7262 (1)	0.5213 (0)	0.7176 (2)	0.4039 (4)
GloVe	L	0.7693	0.6097	0.3884	0.7296	0.2183
BERT	L	0.8124 (2)	**0.7096 (1)**	**0.5191 (0)**	0.7368 (2)	**0.4027 (3)**
LXMERT	LV	0.7821 (27)	0.6000 (27)	0.4438 (21)	0.7417 (33)	0.2443 (21)
UNITER	LV	0.7679 (18)	0.6813 (2)	0.4843 (2)	0.7483 (20)	0.3926 (10)
ViLBERT	LV	0.7927 (20)	0.6204 (14)	0.4729 (16)	0.7714 (26)	0.3875 (14)
VisualBERT	LV	0.7592 (2)	0.6778 (2)	0.4797 (4)	0.7512 (20)	0.3833 (10)
Vokenization	L_V	**0.8456 (9)**	0.6818 (3)	0.4881 (9)	**0.8068 (10)**	0.3439 (9)

FIGURE 3.49: From Pezzelle et al. (2021). Spearman's rank correlation between similarities computed with representations by all tested models and human similarity judgments in the five evaluation benchmarks.

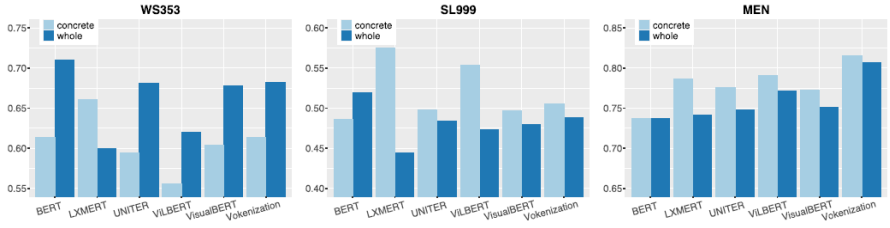


FIGURE 3.50: From Pezzelle et al. (2021). Correlation between model and human similarity ratings on WordSim353, SimLex999 and MEN. Each barplot reports results on both the whole benchmark and the most concrete subset of it.

certain dataset[5] and the whole dataset (thus including more abstract concepts). The results confirm the trend: multimodal models are more effective than pure language models at representing concrete words but in many instances they still lag behind when it comes to more abstract concepts.

Last but not least, few words need to be spent on a topic which has been steadily gaining relevance: **Few-Shot Learning**. To train and test models, a large pool of paired images and texts is often needed and the creation of many of the datasets used in fine-tuning required a huge data collection effort, which had to be performed by human agents. This implies that the creation of such data pools can be very costly. For this reason, there is a growing interest in creating models able to cope with low-resource settings. This boils down to the question: can a model perform well on downstream tasks even with just a *limited number* of training examples? The goal is actually once again, to

[5] See Brysbaert et al. (2014) for information on how *concreteness* of a word can be estimated.

mimic how humans learn: a person does not need to see one thousand pictures of a table, to be able to recognize a table...

	SST-2			QNLI			QQP			MNLI		
Extreme Few-shot	0.1%	0.3%	0.5%	0.1%	0.3%	0.5%	0.1%	0.3%	0.5%	0.1%	0.3%	0.5%
$VOKEN(Bert_{base})$	54.70	77.98	80.73	50.54	51.60	61.96	44.10	60.65	65.46	37.31	54.62	58.79
$iACE(Bert_{base})$	**77.98**	**80.96**	**81.42**	**51.64**	**58.33**	**64.03**	**49.36**	**63.67**	**71.17**	**40.07**	**56.49**	**59.57**
$VOKEN(Roberta_{base})$	70.99	71.10	77.86	54.37	62.23	65.78	62.32	67.25	70.18	48.59	49.76	58.23
$iACE(Roberta_{base})$	**75.34**	**78.66**	**83.60**	**54.79**	**65.03**	**65.83**	**65.43**	**68.11**	**70.77**	**48.94**	**52.74**	**59.39**
Normal Few-shot	1%	3%	5%	1%	3%	5%	1%	3%	5%	1%	3%	5%
$VOKEN(Bert_{base})$	81.40	86.01	84.75	64.17	77.36	80.19	72.55	78.37	80.50	60.45	62.73	72.35
$iACE(Bert_{base})$	**82.45**	**87.04**	**86.47**	**65.09**	**79.54**	**80.52**	**74.31**	**78.69**	**80.52**	**62.15**	**70.43**	**73.73**
$VOKEN(Roberta_{base})$	83.78	84.08	87.61	75.00	81.16	81.23	73.14	79.09	79.63	63.51	70.68	74.02
$iACE(Roberta_{base})$	**83.83**	**84.63**	**89.11**	**79.35**	**81.41**	**81.65**	**73.72**	**79.38**	**79.81**	**65.66**	**70.76**	**74.10**

FIGURE 3.51: From Lu et al. (2022). Model-agnostic improvement in Few-shot Setting with GLUE benchmark.

This table from Lu et al. (2022), where models are trained using only up to 5% of the training set, shows for example the ability for a model supervised with "imagination" (which was a generated visual representation of a certain textual input) to outperform models with only simple visual supervision (the Voken-model). This is just an example, but the ability to perform well in *few-shot* settings has become the touchstone of the evaluation modern multimodal models.

3.3.7 The End Of This Story

We started this story with the *Symbol Grounding Problem*, which affirms that to grasp the meaning of a word, the word has to be put in a context other than the pure linguistic one. We thus investigated some of the architectures proposed to ground words in a visual space in form of static images. The goal (hope) is to better capture the semantics of words, in form of better word embeddings, to be employed in heterogeneous tasks, from *semantic-similarity* to downstream tasks, such as *sentiment analysis*.
From this brief analysis it emerges that grounding words in images can actually improve the representation of *concrete* concepts, whereas visual grounding does not seem to add value to pure language models when it comes to *abstract* concepts. Nonetheless, forms of visual supervision like the *Voken-Classification* task or the employment of generative models which allow to *imagine* words, such as in the *iACE-Framework*, might be the right way to bridge this gap.
The Transformers have been a revolution in the field of NLP and with their advent, the trend has now become to build models with pre-training tasks capable of generating powerful task-agnostic word representations. The knowledge gained with these tasks can be then transferred to downstream tasks with the goal to limit the amount of labeled data necessary to fine-tune models. Labeling data is indeed costly: this is why the ability of a model to generalize

well when exposed to just few training examples has been steadily gaining importance as evaluation metric. This was the so called *few-shot learning*. Moreover, Transformer-based models have "universal vocation": they tend to be multimodal and multi-task, encompassing vision, language and vision and language tasks. This idea might be appealing because humans learn by being exposed to a multitude of different inputs and tasks. But as we have seen, pure language models such as BERT tend to still outperform multimodal multi-task models. There is definitely room for improvement.
One might wonder whether the grounding of words in images is the right way to seek a better representation of words. Well, humans learn using all five senses and maybe the answer might be to incorporate in the models more heterogeneous perceptual information: not only static images but also videos, speech and the like. The debate is still open: the story *goes on*...

Last but not least, a mention needs to be made on concrete applications of these image-empowered word-embeddings. The use of images to support linguistic models has been experimented in several fields, from *Dialogue Response Generation* (e.g. Sun et al. (2021)) to *Machine Translation*, where for example Ive et al. (2019) found images to improve the quality of translation when the textual context is generic and/or ambiguous. The number of potential applications of the models described in this subchapter is growing steadily in the scientific community. But this is yet *another* story...

3.3.8 Appendix: Selected Models - Summary

A table (available here) contains a summary of selected language models augmented with visual components. For each model, the following information are reported:

- Pure language model and pretraining data
- Visual features and pretraining data
- Fusion strategy of the two modalities
- Benchmarks/baselines for evaluation

3.4 Text supporting Vision Models

Author: Max Schneider

Supervisor: Jann Goschenhofer

3.4.1 Introduction

> "The biggest lesson that can be read from 70 years of AI research is that general methods that leverage computation are ultimately the most effective, and by a large margin. [...] Most AI research has been conducted as if the computation available to the agent were constant (in which case leveraging human knowledge would be one of the only ways to improve performance) but, over a slightly longer time than a typical research project, massively more computation inevitably becomes available. Seeking an improvement that makes a difference in the shorter term, researchers seek to leverage their human knowledge of the domain, but the only thing that matters in the long run is the leveraging of computation. [...] One thing that should be learned from the bitter lesson is the great power of general purpose methods, of methods that continue to scale with increased computation even as the available computation becomes very great."
>
> — Sutton (2019)

This insight seems to directly inspire most model choices presented in this chapter. Each network can be seen as an attempt of its creators to employ their vast available resources on a large scale, with a particular focus on dataset sizes. This mostly becomes feasible through the adaptation of recent findings in natural language processing (NLP; see chapter 2.1) to computer vision (CV). On the one hand, architectural concepts firstly popularized in NLP are translated to CV (e.g., self-supervised learning or the Vision Transformer; Dosovitskiy et al., 2020b) (see chapter 2.2). On the other hand, these powerful new NLP models, mostly Transformers (Vaswani et al., 2017b), support bigger models from the inside as text encoding building blocks; hence the name of this chapter. Throughout this chapter, we will introduce recent relevant CV models CLIP (Radford et al., 2021a), ALIGN (Jia et al., 2021b) and Florence (Yuan et al., 2021) and discuss their underlying core concepts. The strong performances confirm the potential, hinted at by the impressive GPT-3 (Brown et al., 2020), of improving CV and increasing scale with the help of NLP.

3.4.2 Concepts

3.4.2.1 Web-scale data

A core problem that troubles researchers is the lack of robustness of previous state-of-the-art CV models to distribution shifts. I.e., when a model with good performance on its original dataset fails to generalize (transfer its knowledge) to new, more or less similar datasets. E.g., Radford et al. (2021a) report that a ResNet101 which they trained on ImageNet to an accuracy of 76.2% maintains only an accuracy of 32.6% on ObjectNet. This suggests that the model perhaps did not learn high quality latent representations, but instead overfit to the dataset-specific data-generating distribution. A common way to tackle this would be to try out various changes on the architecture and the training algorithm of the network. But this kind of adaptation, inscribing expert knowledge into the model, seems to repeat the mistake pointed out by Sutton (2019); "micromanaging" a model is likely to thwart future scaling.

The researchers of CLIP, ALIGN and Florence follow a different approach, based on scale. They try to increase sample size as much as possible and work with tremendous numbers of training observations:

- 400 million (CLIP; Radford et al., 2021a)
- 900 million (Florence; Yuan et al., 2021)
- 1.8 billion (ALIGN; Jia et al., 2021b)

These large-scale dataset are generated using the vast amount of image-text pairs produced by and readily available on the internet. Thus, error prone, cost and labor intensive (difficult to scale), manual labeling is avoided. Unfortunately, the models trained on web data also become vulnerable to their downsides. Because of their extremely noisy nature, still some form of preprocessing is needed, e.g., filtering for English language, excluding graphic content and, optionally, removing images with non-informative alt-texts. This makes some degree of dataset curation, and therefore arbitrary choices, necessary. Likewise, the social biases inherent to the internet are reproduced and furthermore, while this approach improves data efficiency to some degree (see next subsection 3.4.2.2), the poor performance of deep learning in this area is not substantially enhanced and mainly just compensated for with a super scalable source of supervision (Radford et al., 2021a).

3.4.2.2 Contrastive objective

This source of supervision is the information contained in the co-occurrence of the image with its alt-text. It is accessed through natural language supervision. The architectures jointly train two sub-networks for image and text encoding, respectively. During this, the vector encodings are aligned in the latent representation space through minimizing a variant of the contrastive loss function (3.10) (Tian et al., 2020). Half of the first image-text pair loss

$$\ell_1^{V_{\text{img}},V_{\text{txt}}} = - \mathop{\mathbb{E}}_{\{v_{\text{img}}^1, v_{\text{txt}}^1, \ldots, v_{\text{txt}}^N\}} \left(\log \frac{h_\theta(\{v_{\text{img}}^1, v_{\text{txt}}^1\})}{h_\theta(\{v_{\text{img}}^1, v_{\text{txt}}^1\}) + \sum_{k=2}^{N} h_\theta(\{v_{\text{img}}^1, v_{\text{txt}}^k\})} \right), \tag{3.10}$$

where v_{img}^1 and v_{txt}^1 are vector encodings (latent representations) of image 1 and text 1 and $h_\theta(\cdot)$ is a similarity measure. In order to guarantee symmetry, the total loss is formed by the sum of $\ell_1^{V_{\text{img}},V_{\text{txt}}}$ and $\ell_1^{V_{\text{txt}},V_{\text{img}}}$, where the pairwise similarities of one text and every image is calculated instead of the other way around.

Figure 3.52 visualizes this. Initially all images and texts in the training data are encoded by the responsible sub-network. Using the resulting encodings, a similarity matrix with elements $h_\theta(\{v_{\text{img}}^i, v_{\text{txt}}^j\})$ can be calculated. Loosely speaking, the contrastive objective is to maximize elements on the diagonal and minimize the others.

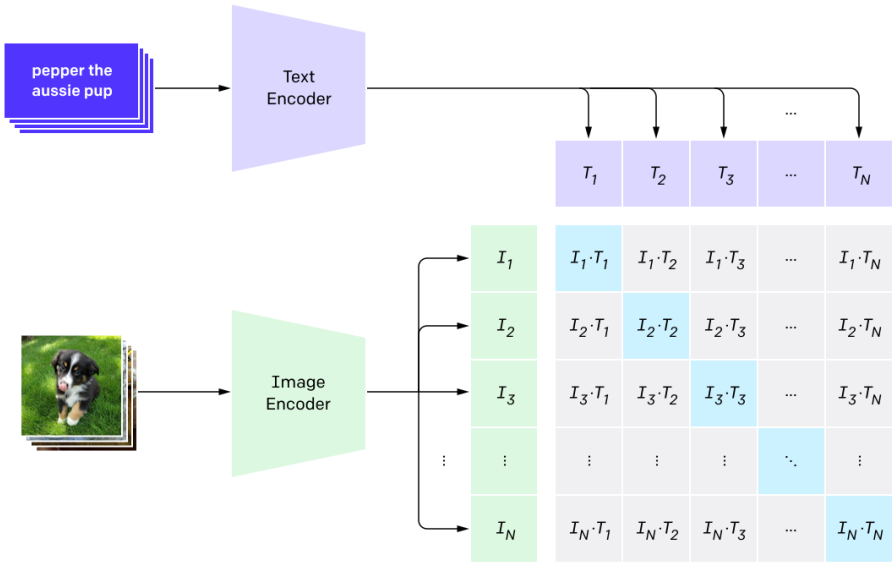


FIGURE 3.52: Visualization of a contrastive objective (Radford et al., 2021a). After encoding the data, a similarity matrix for the images and texts is computed. The aim is that the N true image-text pairs score high in terms of similarity, while the $N^2 - N$ other possible combinations score low.

Contrastive learning can be contrasted with classical predictive learning. Figure 3.53 gives an interesting insight into the choice of space, where goodness of fit is measured. The exemplary task is to color an image given its B/W version. Approach (a) first encodes the B/W image and then decodes the interim latent representation to fitting colors. The goodness of this fit is measured in the output space, meaning the estimated colors are compared to the true colors.

Conversely, approach (b) measures the loss in the representation space.[6] A reason for the good performance of contrastive learning could be that, while common prediction losses (e.g., the $\mathcal{L}_2$ loss) penalize each prediction output dimension independently, approach (b) implies measurement in the intertwined representation space (Tian et al., 2020).

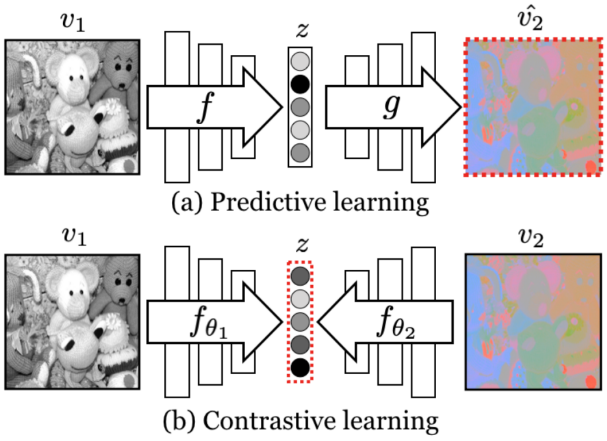


FIGURE 3.53: Predictive vs. contrastive learning: Predictive losses are measured in the output space while contrastive losses are measured is in the representation space, indicated by red dotted boxes (Tian et al., 2020).

But in the end, rather than theoretical considerations, the driving factor for using this objective is data efficiency. As can be seen in figure 3.54, Radford et al. (2021a) start their search for an adequate pre-trained model (more on this in subsection 3.4.2.3) by experimenting with a Transformer-based language model predicting the exact captions of an image. It turns out that this approach trains three times slower, in terms of data efficiency, compared to a simpler baseline of predicting a bag-of-words text encoding. Additionally, switching to the contrastive objective of CLIP improves data efficiency by a factor of four.

Nonetheless, the switch to contrastive learning leads to some limitations. Its rigidity demands certain extra steps and forfeits the high flexibility of generative models. In particular, this means contrastive models similar to CLIP are limited to choose from available options and cannot freely generate texts or images. To extend the capabilities of those models additional network building blocks are necessary.

[6] Note that contrastive learning easily works with other combinations of modalities than text and image; here B/W and colors.

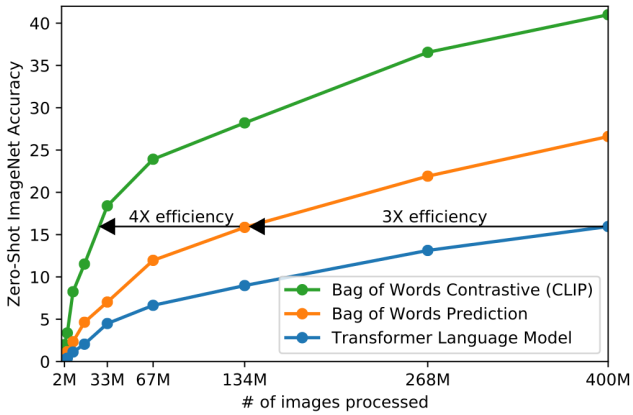


FIGURE 3.54: Data efficiency of contrastive objective. Development of zero-shot accuracy (see next subsection 3.4.2.3) on ImageNet with increasing number of instances of training data processed by the models. The contrastive objective reaches similar accuracy scores as the generative approach with only a seventh of the amount of data (Radford et al., 2021a).

3.4.2.3 Foundation models and zero-shooting

The first models which are considered foundation models today began to appear in NLP. The term, later coined by Bommasani et al. (2021), refers to models that are noteworthy due to their large scale and ability to adapt to a wide variety of downstream tasks. An early example is BERT (Devlin et al., 2018b). Often, foundation models have an unfinished touch to them and the true scope of their capabilities cannot be sketched out clearly. This generally is the case because the desired abilities of neural networks are not designed for explicitly, but rather emerge during their implementation and usage on downstream tasks. Bommasani et al. (2021) cite GPT-3's ability to perform certain types of new tasks solely by confronting it with the right natural language prompt. E.g., it is possible to get GPT-3 to summarize a paragraph by appending "TL;DR" (too long, didn't read) to the prompt, which is a common pattern on the internet to signal a following summery. This is referred to as "in-context learning" (Brown et al., 2020). It is apparent that one can make up plenty of unexpected ways to employ these models and it remains unknown whether there is a further way no one thought of yet. This means possibly saving computational and data collection costs down the line, which ineptly is true for malicious use cases, e.g., surveillance, too.

Foundation models build on the concept of transfer-learning, i.e., pre-training a model on a feasible source task and applying it to the desired downstream task. In the context of this chapter this means pre-training on web-scale data (see subsection 3.4.2.1) and evaluating performance on various common classification datasets. E.g., Radford et al. (2021a) name the SVHN dataset

as a proxy for the task "street number transcription" with the caveat "on the distribution of Google Street View photos", but they remark that a lot of datasets have no obvious, specific task associated, e.g., CIFAR-10. They use these kind of datasets for measuring the "robustness to distribution shift and domain generation" of their model, which still is a topic of great interest as mentioned in subsection 3.4.2.1. When there is no further fine-tuning on the downstream task, i.e., no resuming of training on the new target dataset, this is referred to as zero-shooting. Zero-shooting has the clear advantage of evaluating performance more unbiased, as processes like overfitting to the data-generating distribution will not distort results.

Figure 3.55 shows how contrastive models perform zero-shot transfer. In the case of image classification all available classes are encoded by the language model. Afterwards, the CV sub-network computes the encoding of the image to be classified and all pair-wise similarity scores are returned. The pair with the best score can be retrieved as the decision. Image retrieval works the other way around: After an initial encoding of all images, the ones most similar to the encoded natural language text prompt in the representation space can be returned.

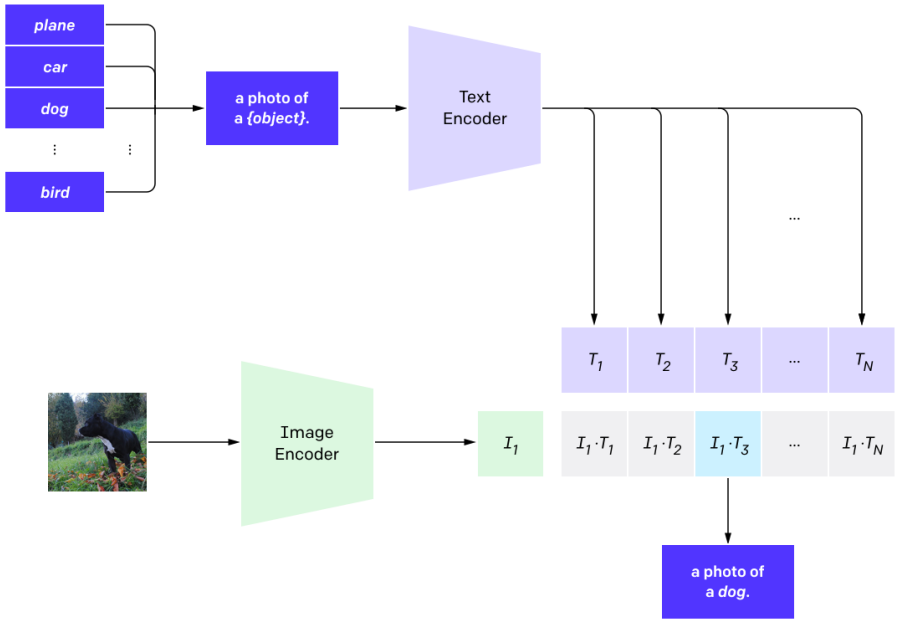


FIGURE 3.55: Visualization of zero-shooting (Radford et al., 2021a).

3.4.3 Architectures

3.4.3.1 CLIP

The first of the large scale contrastive CV models that were published is CLIP, short for Contrastive Language-Image Pre-training (Radford et al., 2021a). The components of its name are explained in previous subsections 3.4.2.2, 3.4.2.1 and 3.4.2.3 and are the crucial concepts of ALIGN and Florence as well. CLIP is a product of OpenAI, but its code is freely available and the different versions can be accessed as python modules. The dataset used for training is not released though.

A lot of preliminary work stems from Zhang et al. (2020b), who introduced contrastive representation learning using image-text pairs. Their implementation of the contrastive loss function (3.10) follows

$$\ell_1^{V_{\text{img}},V_{\text{txt}}} = -\log \frac{\exp(\langle v_{\text{img}}^1, v_{\text{txt}}^1\rangle/\tau)}{\sum_{k=1}^{N}\exp(\langle v_{\text{img}}^1, v_{\text{txt}}^k\rangle/\tau)}, \tag{3.11}$$

where $\langle v_{\text{img}}^1, v_{\text{txt}}^1\rangle$ represents the cosine similarity, i.e., $v_{\text{img}}^{1\top} v_{\text{txt}}^1/(\|v_{\text{img}}^1\|\|v_{\text{txt}}^1\|)$, and $\tau \in \mathbb{R}^+$ is a temperature parameter, which is directly learned during training (Zhang et al., 2020b). CLIP adopts this. $\ell_1^{V_{\text{txt}},V_{\text{img}}}$, the counterpart to $\ell_1^{V_{\text{img}},V_{\text{txt}}}$ for the total loss, is function (3.11) with switched arguments. This can be viewed as a symmetric cross entropy loss over the cosine similarity of the embeddings (Radford et al., 2021a).

Architecture

The text encoder for CLIP (see figure 3.53) is a modified Transformer (Vaswani et al., 2017b), which was also used for GPT-2 (Radford et al., 2019b). For the image encoder multiple sub-networks are evaluated:

- ResNets: ResNet-50, ResNet-101
- ResNets which follow EfficientNet-style model scaling: RN50x4, RN50x16, RN50x64
- Vision Transformers: ViT-B/32, ViT-B/16, ViT-L/14

The best performing sub-network was the ViT-L/14. In turn, they trained it for an additional epoch with higher resolution images (336px), denoting this version ViT-L/14@336px. If not indicated otherwise, the performances of this version of CLIP are displayed. The EfficientNet-style ResNets use x4, x16 and x64 of the compute of a ResNet-50 and the largest model (the RN50x64) trained for 18 days on 592 V100 GPUs, while the ViT-L/14 only took 12 days on 256 GPUs. The high parallelization capabilities of Transformers seem to pay off.

When explaining zero-shooting initially (see subsection 3.4.2.3), a text processing step was skipped. As can be seen in figure 3.55, there is an additional

operation before the labels are fed into the text encoder. In order to help the model understand the context of the words, the class labels are embedded in a sentence, e.g., "A photo of a {label}.". This increases the models zero-shot accuracy on ImageNet by 1.3 percentage points (pp). When ensembling 80 different context prompts[7] Radford et al. (2021a) improve ImageNet accuracy by an additional 3.5pp, which adds up to a total of nearly 5pp. The average performance gain across 36 datasets is reported to be 5pp. It is similarly possible to directly communicate visual concepts like "picture", "macro", "drawing" or even "dog" to the model.

Robustness

Figure 3.56 illustrates the performance of CLIP and a ResNet101, whose training on ImageNet was stopped at the point it reached the same accuracy as zero-shot CLIP. It can be deduced that the methods studied in the paper of Radford et al. (2021a) constitute an important step towards closing the robustness gap mentioned earlier (see subsection 3.4.2.1). While the performance of the ResNet101 deteriorates with datasets generated from more and more different data distributions compared to ImageNet, CLIP remains fairly accurate. Note that these findings have to be taken with a grain of salt. Because OpenAI does not grant public access to their training data, independent parties cannot investigate these claims on their own. E.g., it has to be relied on the conclusions of their overlap analysis to rule out that CLIP has not seen biasing amounts of future test data during training.

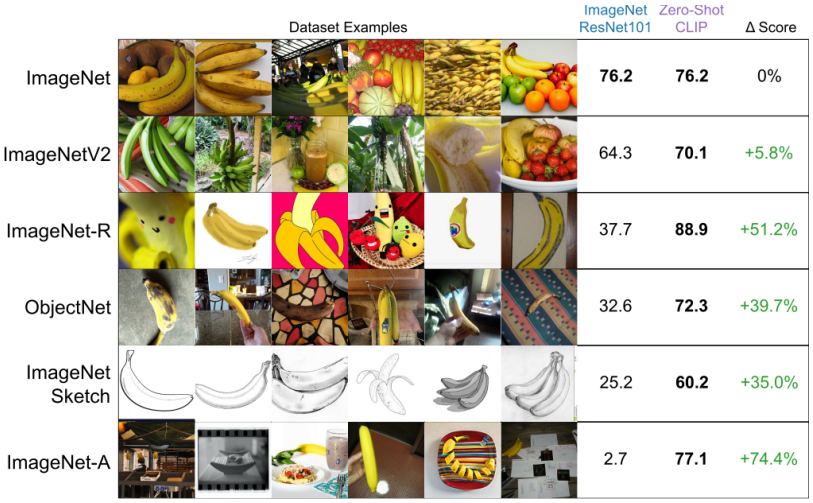


FIGURE 3.56: Robustness of zero-shot CLIP to distribution shifts (Radford et al., 2021a).

[7] Prompts like: "A photo of a big {label}.", "A photo of a small {label}." (Radford et al., 2021a)

CLIP as a building block

Shen et al. (2021) study how the performance of Vision-and-Language (V&L) models improves, when the visual encoder is switched to CLIP's strong image encoder. They discover that in this field of CV the ViT-B scores significantly worse than the ResNets. E.g., tests on image captioning reveal that the V&L model using ViT-B often performs only half as strong as the version using the RN50x4 (the largest network used in this study). This is possibly due to the pooling strategies of ViT-B, which result in a lack of visual localization abilities. Shen et al. (2021) test their hypothesis and generate, e.g., figure 3.57 which depicts Grad-CAM Visualizations for a V&L model with a ViT-B backbone and a ResNet-50 backbone and the question "What color is the woman's shirt on the left?". The red area indicates relevant pixels and appears much more focused for CLIP-Res50 than for CLIP-ViT-B.

(a) Original (b) CLIP-ViT-B (c) CLIP-Res50

FIGURE 3.57: Grad-CAM Visualizations for the prompt "What color is the woman's shirt on the left?".

3.4.3.2 ALIGN

The approach of Jia et al. (2021b) is largely similar to CLIP. They reiterate the necessity of large-scale vision datasets, but assert that even CLIP's data collection process still involves a non-trivial amount of data curation. They propose that the amount of additional observations obtained through minimizing the amount of filtering makes up for the increased noise. Following this rationale, they create a training dataset with 1.8 billion image-text pairs. The corresponding model is named ALIGN, short for "A Large-scale ImaGe and Noisy-text embedding", whose acronym hints at the contrastive loss, which aligns vector encodings in the representation space (see subsection 3.4.2.2).

Architecture

ALIGN follows the dual encoder architecture employed by Zhang et al. (2020b) and Radford et al. (2021a), but uses a part of BERT-Large as the text and EfficientNet-L2 as the image encoder, which they jointly train from scratch. The model has around 800 million parameters (Alford, 2021). Subsection 3.4.4 goes into more detail about the performance of ALIGN and compares all three models discussed in this subsection.

Connecting image and text representations

The contrastive loss function aligns the latent representations of the different modalities. In other words, the explicit objective is that similar vector encodings implicate similar inputs. This means arithmetic operations like the ones mentioned in chapter 2.1 are not only meaningful on encodings belonging to the same modality, but to different modalities. E.g., one can add up the image encoding of a picture of the Eiffel tower and the text encoding of the word "snow" and retrieve pictures with high cosine similarity as a result, see figure 3.58 for an illustration.

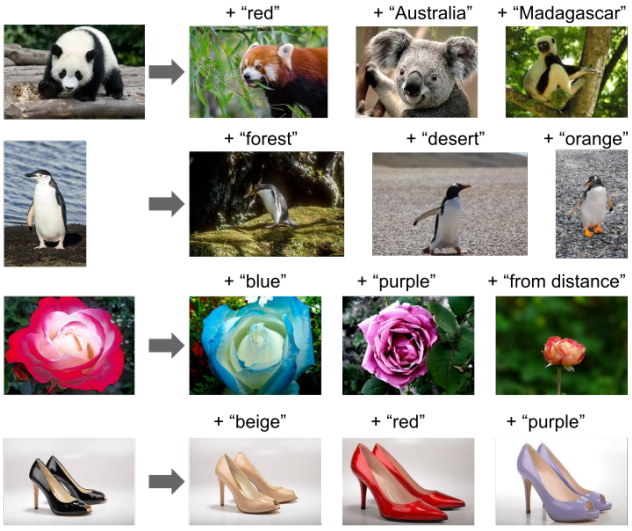


FIGURE 3.58: Multimodal image retrieval via arithmetic operations on word and image embeddings.

3.4.3.3 Florence

While in principle the approach of Yuan et al. (2021) does not largely differ from the others, the focus of this paper is more about creating a true foundation model. In order to achieve this, they propose a map of possible vision applications which the try to cover via extending the core model with modules. As figure 3.59 depicts, they want to advance into the dimensions of fine-grained object detection, dynamic action recognition and true multimodal tasks. Due to their big ambitions, they name their model Florence after "the birthplace of Renaissance" (Yuan et al., 2021).

Architecture

As the two encoders for the pre-trained core they use a hierarchical Vision Transformer (CoSwin Transformer) for images and a Transformer similar to

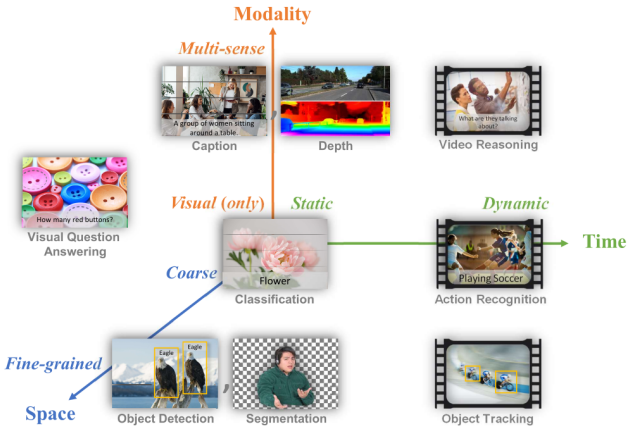


FIGURE 3.59: Florence' approach to foundation models: A general purpose vision system for all tasks.

CLIP's for text. Their 893 million parameters are also jointly trained from scratch on 900 million image-text pairs. The alignment happens in the so called image-label-description space which is encoded through a special version of the contrastive loss function which regards all image-text pairs with the same label as positive instances. Figure 3.60 depicts their version of figure 3.52 where one can schematically see how they flexibly add modules to the pre-trained core in order to adapt to various downstream tasks.

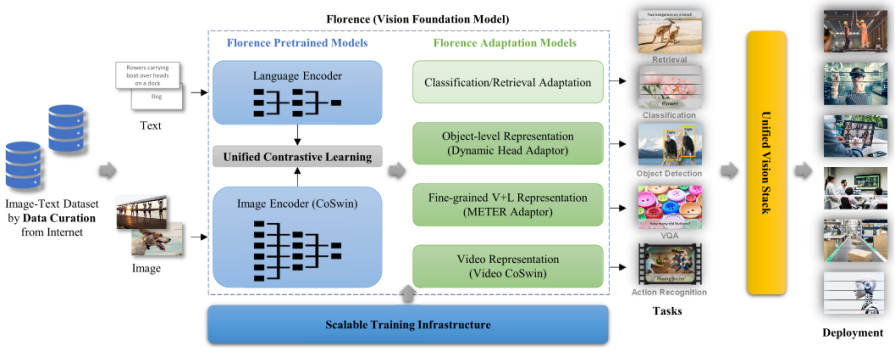


FIGURE 3.60: Modular architecture of Florence.

3.4.4 Performance comparison

Throughout the papers of Radford et al. (2021a), Jia et al. (2021b) and Yuan et al. (2021) we were able to collect three tables with reported performance measures to compare these approaches.

Table 3.61 summarizes the zero-shot accuracies on four different ImageNet variants. Unfortunately Yuan et al. (2021) only stated their performance on the original ImageNet, where they beat CLIP and ALIGN by a margin of 7.3pp. The results on the other three ImageNet pendants are mixed and there is no clear winner between CLIP and ALIGN.

	ImageNet	ImageNet-R	ImageNet-A	ImageNet-V2
CLIP	76.2	88.9	**77.2**	**70.1**
ALIGN	76.4	**92.2**	75.8	**70.1**
Florence	**83.7**			

FIGURE 3.61: Top-1 Accuracy of zero-shot transfer of models to image classification on ImageNet and its variants.

Table 3.62 concerns zero-shot image retrieval on the Flickr30K and the MSCOCO dataset (see chapter 2.3). Even though there are not many major score differences, there is a clear ranking with CLIP on third, ALIGN on second and Florence on the first place.

	Flickr30K (1K test set)				MSCOCO (5K test set)			
	Image→Text		Text→Image		Image→Text		Text→Image	
	R@1	R@5	R@1	R@5	R@1	R@5	R@1	R@5
CLIP	88.0	98.7	68.7	90.6	58.4	81.5	37.8	62.4
ALIGN	88.6	98.7	75.7	**93.8**	58.6	83.0	45.6	69.8
Florence	**90.9**	**99.1**	**76.7**	93.6	**64.7**	**85.9**	**47.2**	**71.4**

FIGURE 3.62: Zero-shot image and text retrieval (Yuan et al., 2021).

The most comprehensive comparison is shown in table 3.63. It depicts the accuracy of zero-shot CLIP and Florence on various datasets as well as the scores of all three models fine tuned to the respective datasets. Florence beats CLIP in nearly all evaluations, for the zero-shot setting as well as for fine tuned performance. Jia et al. (2021b) only report on four of these twelve datasets, where they win half of the time.

Summing up, ALIGN achieves its goal of replicating CLIP's impressive performance while dramatically reducing the required data curation effort and Florence has the overall top performance. This could be attributed to its custom loss, maybe to Yuan et al. (2021) striking the best balance between sample size and data curation or to Florence having the best sub-networks; or a combination of all three.

Once again note that none of the training datasets were made publicly available. It cannot be guaranteed that all benchmarks were evaluated on unseen datasets.

	Food101	CIFAR10	CIFAR100	SUN397	Stanford Cars	FGCV Aircraft	VOC2007	DTD	Oxford Pets	Caltech101	Flowers102	ImageNet
CLIP	93.8	95.7	77.5	68.4	78.8	37.2	84.3	55.7	93.5	92.8	78.3	76.2
Florence	95.1	94.6	77.6	77.0	93.2	55.5	85.5	66.4	95.9	94.7	86.2	83.7
CLIP (fine tuned)	95.9	**97.9**	**87.4**	82.2	91.5	71.6	89.9	83.0	95.1	96.0	99.2	85.4
ALIGN (fine tuned)	95.9				**96.1**				**96.2**			88.6
Florence (fine tuned)	**96.2**	97.6	87.1	**84.2**	95.7	**83.9**	**90.5**	**86.0**	96.4	**96.6**	**99.7**	**90.1**

FIGURE 3.63: Top-1 Accuracy of CLIP, Florence and ALIGN on various datasets.

3.4.5 Resources

One can access the pre-trained CLIP models on Github and they even found their way into simple command line tools already. For example there is a CLI named rclip, which can be used for personal image retrieval, wrapping the *ViT-B/32* CLIP architecture. On a (mid-range, regular) laptop, we were able to find seemingly good matches for search terms which we tried out inside a folder containing about 100 different pictures. After an initial caching one request took about ten seconds. Furthermore CLIP continues to be used inside new models, e.g., DALL·E 2, where it is used for the image embedding (Ramesh et al., 2022b). Also, there is a crowd-sourcing effort to replicate CLIP's training dataset called LAION-400M (Schuhmann, 2022). To validate the image-text pairs collected for this, their cosine similarity is computed using CLIP and instances with a value too low are discarded. To our knowledge no resources were open-sourced as part of the other two papers ALIGN and FLORENCE.

3.5 Models for both modalities

Author: Steffen Jauch-Walser

Supervisor: Daniel Schalk

Data is naturally at the heart of every data scientific issue. While there have been many advances made in machine learning in recent years, many promising research areas remain, as do a multitude of problems associated with them. One such promising area are multi-modal machine learning models. Combining different input data is a key aspect towards making models more sophisticated. When thinking about teaching robots specific tasks, detecting hateful memes or deep fakes, it is apparent that only through the combination of multiple modalities, success might be achieved. Context is key.

However, learning context requires increasingly complex models. While early

machine learning models built their success upon the possibility to analyze the big pool of available, often unstructured data, modern machine learning models are so demanding that there is often not enough data or training time available. Obtaining data is a major issue for multi-modal machine learning. Since labelling data in vast amounts is prohibitively expensive, larger models have to come up with specific strategies to move forward such as self-supervised training or automatically scraped web datasets. Nevertheless, when models become so large that billions of parameters have to be learned, even scraping the whole web starts to show its limits. Another natural issue is the transformation of different types of data into usable model inputs.

There is no shortage of different single modality machine learning models. On the contrary, when every new hyperparameter configuration might be seen a new model, it becomes hard to keep track. More importantly, it is often not clear how a model from one area transfers to another. Did we learn some modality specific bias or a general principle? Consolidating different models into a unifying framework is a key prospect of multimodal machine learning. While the grand dream of a single unifying model might be out of reach, consolidating different areas is well in sight. In the following, we will have a look at the challenges and prospects of multimodal machine learning against the background of visual language models. Visual Language Models are models which can deal with both language and images as input data. Specifically, we will have a closer look at three different models: Data2vec, VilBert and Flamingo. Data2vec is an unsupervised model that can handle different modalities, but not their interaction, using a single unifying training framework. VilBert is an early visual-language model that can handle interactions between images and text through its innovative concept of cross-attention. Flamingo is a recent few shot visual language model that features large expressive text capabilities through the use of a large language model. With 80B parameters, it particularly highlights how to leverage the communication between frozen models when further scaling up the model size.

An overview across the popularity of current research fields in visual language modelling is provided in figure 3.64. A detailed list of trends for each of those fields can be found in Uppal et al. (2022). Most research is done in the areas of visual question answering (VQA) and visual captioning (VC), but also for example visual commonsense reasoning (VCR), vision-language navigation (VLN) or multimodal affective computing (MAC). MAC uses images and text to infer sentiment, for example through facial expressions. VCR as an extension of VQA is particularly interesting in the realm of making models more interpretable. After all, we would like to know why machine learning models do what they do. Finally, VLN has many promising practical applications in the field of robotics, particularly the interaction of humans and robots.

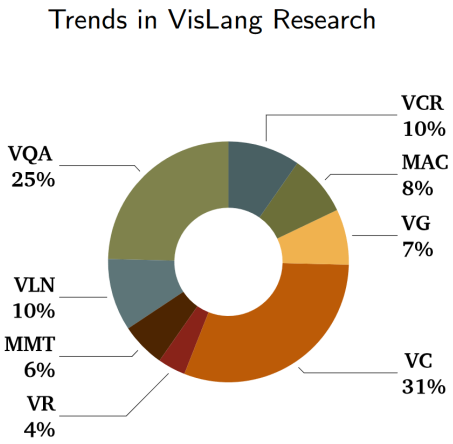


FIGURE 3.64: Uppal et al. (2022): VisLang Paper Trends (previous 2 years)

3.5.1 Data2vec

With data2vec (Baevski et al., 2022), data scientists at Meta, formerly Facebook, developed an architecture that addresses some of the mentioned issues while highlighting the importance of sophisticated training schemes. Their algorithmic structure is able to work with either text, image or speech data. On top of that, the model is self-supervised based on a teacher-student relationship which reduces the need for human labelling. It is not a universal model in the sense that it works with any input, nor is it even a general model in the sense that the algorithm is exactly the same for each modality. However, the overall model structure remains the same for either text, speech or image input data, while only the specific encoding, normalization and masking strategies are modality-specific. In that regard, it is a step towards a more general way of dealing with different modalities and it is very effective at doing so given the benchmark results on typical data sets. Particularly noteworthy is also the way they implement the self-supervised learning. Data2vec predicts contextualized and continuous representations rather than typically used discrete tokens such as sub-words. Working with latent representations of the input space has two advantages: not only is the number of prediction targets not a-priori limited, but they are also richer in information.

Figure 3.65 depicts the general model architecture. The two main components are a teacher and a student model which only differ in one aspect, the weights of the teacher model are an exponentially decaying average of the student's weights. The purpose of the teacher model is to create training targets for the student model. In a first step, a modality is chosen and inputs are encoded according to the specific encoding scheme for that modality. A masked version

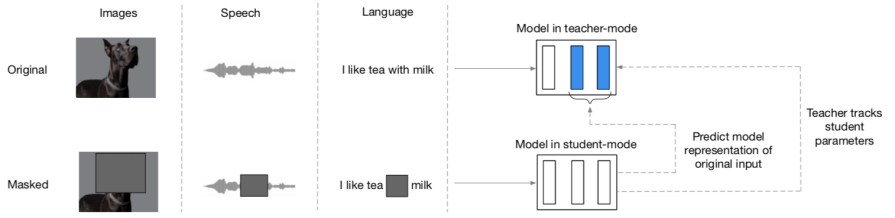


FIGURE 3.65: Baevski et al. (2022): Data2vec Architecture - a teacher model creates contextualized latent targets on the basis of its top K layers (blue) as prediction task to train the student model

is given to the student model, but notably, the teacher model has access to an unmasked, complete view of the input data. Hence, the resulting training targets will be fully contextualized using a self-attention mechanism over the whole input data. The training targets are based on the top K layers of the teacher model depicted in blue in Figure 3.65. More specifically, denoted by y_t, the training target at time t and by a_t^l the outputs of the l-th block, then $y_t = \frac{1}{K}\sum_{l=L-K+1}^{L} \hat{a}_t^l$, i.e. the training targets are the average of the outputs of the top K layers of the teacher network after a normalization has been applied. Normalization helps to stabilize the training process and prevent model collapse which can be an issue with models that learn their own representation.

From the authors point of view, working with a latent representation of the actual learner as training target is a simplification of many commonly used modality-specific designs despite the caveat that this paper still uses modality-specific encoding strategies. Compared to other models, there is no cross-modality training. The specific loss function used to regress the targets is a smooth L1 loss.

$$L(y_t, f_t(x)) = \begin{cases} \frac{(y_t - f_t(x))^2}{\beta} & \text{if} \quad |(y_t - f_t(x))| \leq \beta \\ |(y_t - f_t(x)| - \frac{\beta}{2} & \text{otherwise} \end{cases}$$

Using a smooth L1 loss has the advantage of being continuous, yet sensitive to outliers, however the β parameter needs tuning. As far as the general model architecture is concerned, the underlying architecture is a standard transformer architecture (Vaswani et al., 2017b).

How does the modality specific input handling work?

In many ways, in this work the authors combine the strategies developed in multiple previous works and add a unifying framework on top of it. For images, the typical Vision Transformer (ViT) strategy (3.66) to transform images with a size of 224x224 pixels into 16x16 pixel patches is employed. Every patch is then linearly transformed into a sequence of 196 flattened

representations including a learn-able positional encoding that serve as input to the vision transformer. A classification token is used to produce the final categorization. The contextualization is produced in the multi-head attention blocks as explained in earlier chapters. In short, multi-head attention first projects the keys, queries and values with learned linear projections which are then evaluated in parallel to create more expressive attention maps. Attention itself is calculated as scaled dot-product-attention using a softmax over the scaled product of keys, queries and values (Vaswani et al., 2017b). As far as the vision transformer itself is concerned, datav2vec tests two different model sizes, a base model size of 12 and a large model of 24 transformer blocks. The masking strategy for images follows the Bert pre-training approach of image transformers, BEiT, proposed by Bao et al. (2021). In particular, multiple adjacent blocks are being masked with random aspect ratio. The minimum size of a masked block is 16 patches. In total, 60% of patches were masked in the data2vec algorithm, which is an increase over the original 40% used by BEiT. However, the authors note that they found increased masking to be more accurate. The augmentation strategies are similar, as well. Resizing crops, horizontal flipping and colour jittering were used. Naturally, the student and teacher model are the given the same modified image. Finally, for image data, the model is measured on a classification task. Hence, the authors use a mean-pooling over all patches in the last transformer block and input that into a softmax-normalized projection that conducts the classification, which is again based on the BEiT model.

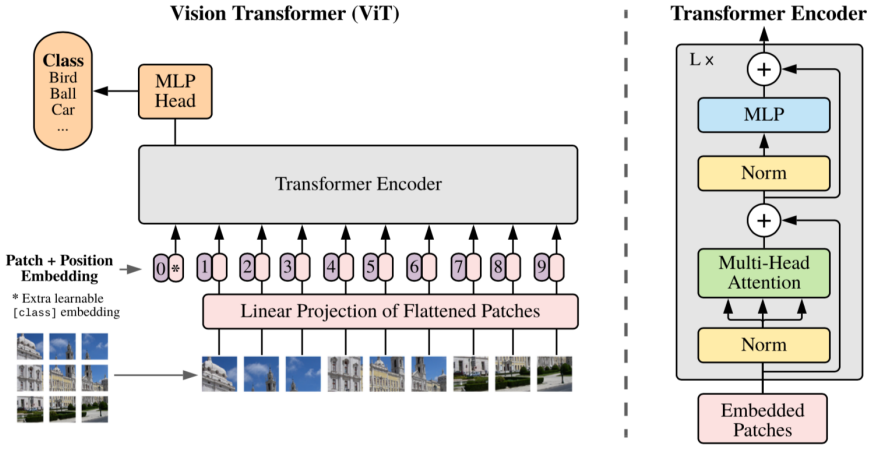


FIGURE 3.66: Dosovitskiy et al. (2021)

The natural language processing model is implemented with a PyTorch toolkit named fairseq and based on the RoBERTa (Liu et al., 2019b) architecture which redesigned the standard Bert model training procedure to make it more

robust and effective. In particular, it increases hyperparameters such as the learning rate and the batch size. It also removes the next sentence prediction task to improve on the masked language modelling performance. In this case they follow Sennrich et al. (2015b) and encode sub-words as 50k byte-pairs. A separate embedding vector is learned for each type. For the masking, the Bert masking is being used. 15% of the embedded tokens are replaced, thereof 80 percent are learned masks, 10% are unchanged and the remaining 10% are replaced with random tokens in the vocabulary. Another strategy that the authors also consider is the wave2vec masking strategy' to mask four consecutive tokens with a probability of 0.35 while only using learned tokens (Baevski et al., 2020). As it turns out, the later strategy further improves the results. The natural language processing model is evaluated on the General Language Understanding Evaluation (GLUE) benchmark (Wang et al., 2018) which includes for example includes NLP inference, sentence similarity and sentiment analysis tasks.

The speech category is also implemented in fairseq. The feature encoder for speech is based on the wave2vec framework and uses 16 kHz inputs. It is built upon seven temporal convolutions intertwined with normalization layers and a GELU activation function such that the output of the encoder is 50 kHz.

As far as the results are concerend, data2vec achieved state-of-the-art performance in vision and language tasks among similar self-supervised models.

	ViT-B	ViT-L
Multiple models		
BEiT (Bao et al., 2021)	83.2	85.2
PeCo (Dong et al., 2022)	84.5	86.5
Single models		
MoCo v3 (Chen et al., 2021b)	83.2	84.1
DINO (Caron et al., 2021)	82.8	-
MAE (He et al., 2021)	83.6	85.9
SimMIM (Xie et al., 2021)	83.8	-
iBOT (Zhou et al., 2021)	83.8	-
MaskFeat (Wei et al., 2021)	84.0	85.7
data2vec	84.2	86.6

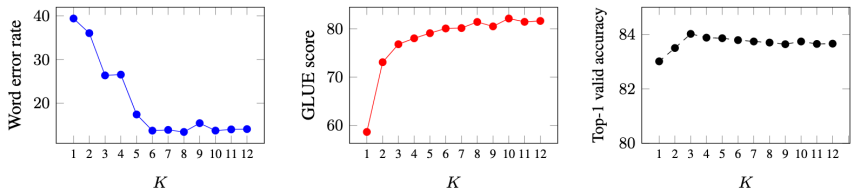
FIGURE 3.67: Baevski et al. (2022): data2vec performance (vision)

Figure 3.67 shows the model's performance in computer vision. Pre-trained and fine-tuned simply on the data of the well known ImageNet-1K dataset, data2vec was evaluated using top1-accuracy, the standard notion of accuracy, on the task to predict single labels for images. The base model ViT-B comprises 86M parameters and ViT-L 307M parameters. The results show that predicting contextualized latent representations in a masked prediction setup can work well as model training compared to classical local methods such as predicting visual tokens. MoCov3 (Chen et al., 2021) is a self-supervised model trained

on a contrastive loss. The most similar model is DINO (Caron et al., 2021), which also uses a self-distillation setup to predict teacher outputs using a cross-entropy loss. However, their prediction target was the final layer rather than averaged layers while using differing images for teacher and student network. The well performing MAE model (He et al., 2022) is a masked autoencoder which is trained on reconstructing masked pixels using an asymmetric encoder-decoder architecture. In contrast, MaskFeat (Wei et al., 2022) uses masked feature prediction. Notably, data2vec outperforms all of them although trained for the same amount or less. Particularly, MAE and MaskFeat use 1600 epochs rather than 800 like data2vec.

Table 3. Natural language processing: GLUE results on the development set for single-task fine-tuning of individual models. For MNLI we report accuracy on both the matched and unmatched dev sets, for MRPC and QQP, we report the unweighted average of accuracy and F1, for STS-B the unweighted average of Pearson and Spearman correlation, for CoLA we report Matthews correlation and for all other tasks we report accuracy. BERT Base results are from Wu et al. (2020) and our baseline is RoBERTa re-trained in a similar setup as BERT. We also report results with wav2vec 2.0 style masking of spans of four BPE tokens with no unmasked tokens or random targets.

	MNLI	QNLI	RTE	MRPC	QQP	STS-B	CoLA	SST	Avg.
BERT (Devlin et al., 2019)	84.0/84.4	89.0	61.0	86.3	89.1	89.5	57.3	93.0	80.7
Baseline (Liu et al., 2019)	84.1/83.9	90.4	69.3	89.0	89.3	88.9	56.8	92.3	82.5
data2vec	83.2/83.0	90.9	67.0	90.2	89.1	87.2	62.2	91.8	82.7
+ wav2vec 2.0 masking	82.8/83.4	91.1	69.9	90.0	89.0	87.7	60.3	92.4	82.9



(a) Speech (b) NLP (c) Vision

Figure 2. Predicting targets which are the average of multiple layers is more robust than predicting only the top most layer ($K = 1$) for most modalities. We show the performance of predicting the average of K teacher layer representations (§3.3). The effect is very pronounced for speech and NLP while for vision there is still a slight advantage of predicting more than a single layer.

FIGURE 3.68: (res:data2vecresults2)

(res:data2vecresults2) Baevski et al. (2022): data2vec results (language)

Figure 3.68 shows the performance in the language domain. For the language domain, the model is evaluated on the GLUE benchmark (Wang et al., 2018). The model is pre-trained and fine-tuned separately on the labelled data from each task. Accuracy is reported as the average across 5 tuning cycles. While data2vec achieves a higher average performance than the baseline model, there are tasks where the baseline model prevails. A large portion of the performance difference seems to be driven by the CoLA task. The Corpus of Linguistic Acceptability (CoLA) consists of 10657 sentences from 23 linguistics publications and the task is to judge whether they are grammatically correct. Hence, it is distinctly different from the other tasks. The Stanford Sentiment Treebank (SST) analyzes sentiment in language through movie reviews. The Multi-Genre Natural Language Inference (MultiNLI) corpus contains sentence

pairs and focusses on textual entailment across genres. Similar tasks are used in the Recognizing Textual Entailment (RTE) dataset which focuses on text from news and Wikipedia. The QNLI (Question-answering NLI) dataset is a Natural Language Inference dataset that contains answers from Wikipedia to corresponding questions posed by an annotator. The task for the model is to find out whether the sentence contains the answer to the question. QQP stands for Quora Question Pairs, which analyzes paraphrases. Finally, the Microsoft Research Paraphrase Corpus (MRPC) also consists of sentence pairs from newswires which may or may not be paraphrases of each other.

As a suitable baseline model, the authors retrain RoBERTa in the respective setup. On top of the heterogeneous performance across language tasks, the evaluation also clearly shows that averaging over multiple layers to create prediction targets improves performance across all three domains. The effects seem to be most pronounced on NLP tasks whereas CV does not benefit from averaging more than three layers. In the speech domain, six layers seems to be enough to reach peak performance. In any case, performance loss while following the strategy to simply average the maximum amount of layers, rather than fine-tuning K, seems small enough to be potentially acceptable.

To sum it up, data2vec is a self-supervised model that can work with either text, speech or image data, but not across modalities. It aims at unifying the learning framework through a teacher-student-setup that allows for contextualized latent target prediction. The teacher model is based on a complete view of the input data, which introduces contextualization, while the student model only sees a masked version of the input. Compared to previous work, the authors average the top K layers rather than only the final layer of the model, which has a notable effect as shown in 3.68. As there are different layers in the transformer network, the authors also investigate which layers work best for prediction. They conclude that the output of the feedforward layer works best. Built on a transformer architecture, self-attention is the main driver that creates contextualized targets in the teacher model and hence performance. The authors also show that contextualization through the teacher model works best with the complete view of the input rather than a partial view. On top of not being able to work across modalities, one drawback is that the model's structure still uses modality specific encoding and masking schemes. In that regard, the perceiver architecture (Jaegle et al., 2021a) for example used in the Flamingo model is a complementary approach worth exploring. An earlier model that works across modalities is VilBert.

3.5.2 Vision-and-Language Bert (VilBert)

As seen in the previous section, data2vec can handle text, image or speech as input data. However, it cannot do so at the same time. The model's focus is on unifying the training approach rather than working across modalities. However, when we think about multimodal models, we usually think of working with

different modalities at the same time. VilBert (Lu et al., 2019b) is a natural extension of the iconic Bert architecture (Devlin et al., 2018c) to vision-and-language modelling. An immediate question is whether vision and language inputs should be handled together in a single stream or in parallel. As we will see, it turns out that encoding inputs in parallel and working with parallel streams increases performance. At heart of that architecture is a co-attention mechanism which enables information exchange between both modalities.

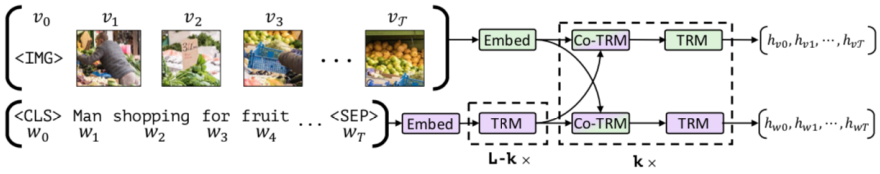
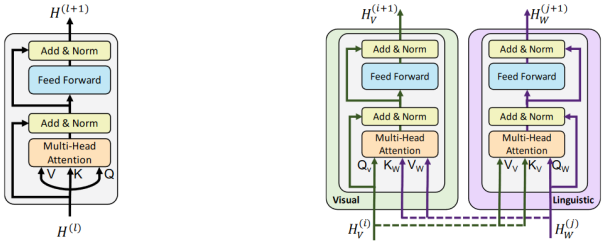


FIGURE 3.69: Lu et al. (2019b): VilBert's Dual Stream Architecture: dashed transformer modules can be repeated, co-attention modules allow sparse interaction between modalities.

Figure 3.69 shows the employed parallel stream architecture. Each modality is handled separately and fed into two Bert-style transformer models. This allows for both modalities to be handled according to their respective needs while co-attention layers allow for communication between the streams. For the language stream, the encoding uses the vocabulary plus a special classification token (cls), a sentence separation token (sep) and a masking token (mask). For the vision stream, image region features are extracted via a Faster R-CNN (Ren et al., 2015) model which was pre-trained on the Visual Genome Dataset (Krishna et al., 2016). Since image regions lack a natural ordering, its spatial location has to be encoded, as well. VilBert achieves that through a five dimensional vector that encapsulates the image coordinates and the fraction of the covered image area. Through projection, the dimensions of the positional encoding and visual features are matched and then summed. The image token marks the beginning of such an image region sequence while representing the whole image.

Through the dual stream architecture, the complexity of the model can be adjusted separately for each modality. An alternative approach would have to discretize the visual space via clustering and then use the resulting tokens in the same way as text tokens. The drawbacks of that approach are the potential loss of detail at the discretization stage and the loss of flexibility across modalities as a result of the same processing. Finally, a single stream architecture can interfere with the pre-training of the language models. The model will have to be fine-tuned based on the created visual tokens. As those might be very different from the text tokens, there is potential for the pre-trained language model to be become 'damaged' in the process and lose capabilities - and idea that is also central to the Flamingo model presented later on.



(a) Standard encoder transformer block (b) Our co-attention transformer layer

Figure 2: We introduce a novel co-attention mechanism based on the transformer architecture. By exchanging key-value pairs in multi-headed attention, this structure enables vision-attended language features to be incorporated into visual representations (and vice versa).

FIGURE 3.70: Lu et al. (2019b): Cross-Attention in VilBert

The key innovation in the Vilbert paper (Lu et al., 2019b) is the use of co-attention layers. In figure 3.70, the basic architecture is depicted. The co-attention module computes query, key and value matrices in a standard transformer attention fashion. However, it then feeds the keys and values from each modality into the other modalities multi-head-attention block. As a result, the visual attention will be conditioned on text whereas the language attention will be image-conditioned. This communication between streams only occurs at specific sections in the model, denoted by co-trm in figure 3.69. Notably, the language stream features a lot more preprocessing before the first co-attention layer than the image stream.

An interesting question to ask is what is actually learned in those attention layers and how they correspond to human attention maps. (Sikarwar and Kreiman, 2022) analyze the efficacy of co-attention layers for VQA tasks in a VilBert network. Specifically, they compute the question conditioned image attention scores and compare them to human attention maps created in experiments. In those experiments, humans are tasked with unblurring specific image regions to answer the same questions one would expect the machine learning model to answer. Such human attention maps are collected in the VQA-HAT dataset (Das et al., 2017). Rank correlation is used to compare attention maps. Sikarwar and Kreiman (2022) find that in a 6 layer network rank correlation plateaus at layer 4 and increases in the number of image regions proposed while encoding the images. Perhaps more surprisingly, they find a minimal influence of semantics on the generation of the attention maps. Randomly shuffling words in a sentence when testing the model performance barely changes the attention output, which suggests that keywords rather than sentence structures drive the attention output. Note however that while attention maps remained similar, the model's actual performance on answering the questions dropped notably by approximately 15% such that it seems clear

that coherent sentences are important for the overall VQA task, but not for the attention creation process. What are the keyword that drive cross-attention in VilBert? The evidence provided by the authors clearly shows that nouns are the most influential parts-of-speech when considering attention maps. On top of that, prepositions can sometimes help identify spatial relations. There is also some support for the hypothesis that removing Wh-words such as "who" and "where" can improve fine-grained attention maps in the final layer which might be worth exploring further as preprocessing for deeper networks. Another approach would be to search for ways to improve the way attention maps are generated by finding ways to include more of the available sentence information. Most notably, however, using object-based region proposals to process images can lead to bottlenecks that can prevent the model from learning sufficiently fine-grained attention maps as shown in figure 3.71. Overall, humans are naturally good at VQA tasks. Hence, it is not surprising that attention maps which correlate well with human attention maps also improve model performance.

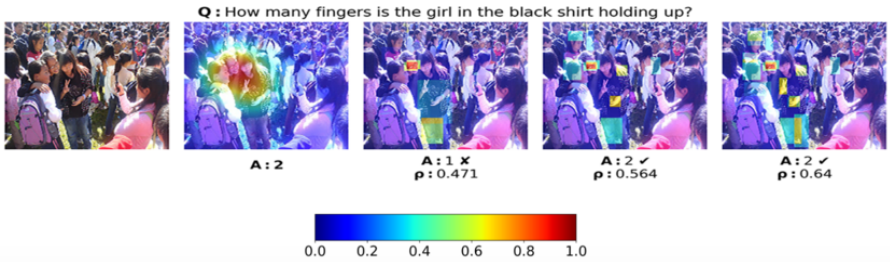


FIGURE 3.71: Sikarwar and Kreiman (2022): (Left to Right) Picture, Human Attention, 36 Regions, 72 Regions, 108 Regions. Similarity between human and model attention is measured using rank correlation.

Figure 3.71 shows that the number of region proposals fed into the model after processing an image affects the ability of the model to produce adequate attention maps. In this particular case the question "How many fingers is the girl in the black shirt holding up?" was correctly answered by humans, as well as a VilBert model using 72 or 108 region proposals. It was answered incorrectly when using only 36 region proposals. Note however that in either case, the machine learning model captured the face of the wrong girl. The model using 72 regions also identified the wrong hand despite answering the question correctly. While the 108 region model identifies the correct hand holding up the fingers, it does not seem to prioritize it over the other identified hands in the picture. Hence, the attention maps are sufficiently different from the human attention map which highlights the need to look closer not only at how models are performing, but also into how their performance has been achieved.

As far as the model training is concerned, VilBert is pre-trained and fine-tuned.

The pre-training tasks comprise masked-multi-modal modelling and multi-modal alignment prediction performed on the Conceptual Captions dataset. That dataset contains about 3,1 million usable aligned image-caption pairs, which have been automatically scraped from web images. For the alignment task, the authors create unaligned images by randomly mismatching captions and images. For the masking task, 15% of the both the visual and language tokens are masked. The task is to reconstruct the mask from the remaining input in a classical Bert fashion. While the text masks are directly regressed like in Bert, the model predicts distributions over semantic classes for the image regions. This is achieved through minimizing the KL divergence, a measure for the similarity of distributions, between the output distribution of the pre-trained model used in feature extraction and the VilBert predictions.

The performance results are depicted in figure 3.72.

Table 1: Transfer task results for our ViLBERT model compared with existing state-of-the-art and sensible architectural ablations. † indicates models without pretraining on Conceptual Captions. For VCR and VQA which have private test sets, we report test results (in parentheses) only for our full model. Our full ViLBERT model outperforms task-specific state-of-the-art models across all tasks.

		VQA [3]	VCR [25]			RefCOCO+ [32]			Image Retrieval [26]			ZS Image Retrieval		
	Method	test-dev (test-std)	Q→A	QA→R	Q→AR	val	testA	testB	R1	R5	R10	R1	R5	R10
SOTA	DFAF [36]	70.22 (70.34)	-	-	-	-	-	-	-	-	-	-	-	-
	R2C [25]	-	63.8 (65.1)	67.2 (67.3)	43.1 (44.0)	-	-	-	-	-	-	-	-	-
	MAttNet [33]	-	-	-	-	65.33	71.62	56.02	-	-	-	-	-	-
	SCAN [35]	-	-	-	-	-	-	-	48.60	77.70	85.20	-	-	-
Ours	Single-Stream†	65.90	68.15	68.89	47.27	65.64	72.02	56.04	-	-	-	-	-	-
	Single-Stream	68.85	71.09	73.93	52.73	69.21	75.32	61.02	-	-	-	-	-	-
	ViLBERT†	68.93	69.26	71.01	49.48	68.61	75.97	58.44	45.50	76.78	85.02	0.00	0.00	0.00
	ViLBERT	**70.55 (70.92)**	**72.42 (73.3)**	**74.47 (74.6)**	**54.04 (54.8)**	**72.34**	**78.52**	**62.61**	**58.20**	**84.90**	**91.52**	**31.86**	**61.12**	**72.80**

FIGURE 3.72: Lu et al. (2019b): VilBert Performance

As mentioned before, the dual stream architecture outperforms the single stream architecture. Furthermore, pre-training considerably boosts performance, as does fine-tuning. Interestingly, the authors also study the effect of the size of the dataset and effect of the architecture depth. Performance increases monotonically with dataset size, suggesting that performance can be further improved with more data. The results on the optimal layer depth are task dependent. VQA and Image Retrieval reach peak performance at 6 layers, where a layer denotes a repeatable block as depicted in figure 3.69. Zero Shot Image retrieval greatly benefits from even deeper depth. However, the VCR and RefCOCO+ tasks seemingly benefit from shallower models. The VQA task is based on the VQA 2.0 dataset. Each image must be matched to one of ten answers. Hence, the VQA task is not open-ended, but treated like a classification task. To achieve that, the model is amended by two MLP layers which use the element-wise product of the model-generated img and cls tokens. The VCR task is also posed as a multiple choice problem with images from movie scenes. To fine-tune for the task, questions and answers are concatenated into four different text input and given as model input together

with the image. In the end, four scores are generated accordingly and selected through softmax. The RefCoCO+ task is a grounding task. An image region has to be selected according to a natural language reference. Caption-Based Image Retrieval requires the model to find an image that corresponds to a selected caption. The dataset used is the Flickr30k dataset which contains 30 000 pictures with five captions that are of higher quality than the automatically generated captions from web data.

3.5.3 Flamingo

The VilBert model showed one way how to actually combine visual and language inputs. In contrast, data2vec showed how to design an unsupervised model and how influential the actual training process as well as contextualization can be. A natural question to ask is then is whether we can build a truly multimodal architecture like VilBert that is self-supervised like data2vec or at little task-specific training and how to optimized its training procedure. In particular, both VilBert and data2vec were tested on multiple tasks, but each task needs slight re-adjustments to the model as well as additional fine-tuning. Ideally, a multimodal architecture would not only be efficient in its initial training, but also easily adaptable to different tasks. Finding ways to not only work with different input modalities, but also with different task is crucial towards building a more general AI. A promising approach in that direction is few shot learning. The following section presents Flamingo (Alayrac et al., 2022), a few shot multimodal architecture developed by Google which comprises key innovations such as handling arbitrarily interleaved vislang sequences as inputs, as well as ways to effectively combine pre-trained vision-only and language-only models. As such, it is a visually conditioned autoregressive text generation model.

Figure 3.73 demonstrates Flamingos capabilities. It can function as chat bot, describe pictures, work with image sequences (videos) and in doing so, simply needs a few prompts.

At the heart of the model is a large language model, Chinchilla (Hoffmann et al., 2022), with 70B parameters. Large language models such as GPT-3 (Brown et al., 2020), as their name suggests, can be trained on a large amount of text data which gives them impressive text generative capabilities. However, multimodal generative modelling presents some specific challenges not present in language-only modelling. First of all, training large language models is expensive. Hence, it is paramount to work with a pre-trained version, but trying to teach a large language model the means to work with visual inputs, as well, has the potential to deteriorate or destabilize the pre-trained model. Second, large language models can suffer from memory constraints that are potentially severely aggravated by simply adding high-dimensional visual data into an input sequence. Third, good generalist capabilities typically require a huge amount of heterogeneous training data. There might not exist enough

FIGURE 3.73: Alayrac et al. (2022): Flamingo Prompt-Output-Examples

labelled image-caption-pair data to successfully accomplish training a capable few shot learning model in the vision-and-language domain. To train Flamingo, the authors solve these challenges by foremost exploring ways to generate their own web-scraped multimodal data set similar to existing ones in the language-only domain. Furthermore, they use a perceiver architecture (Jaegle et al., 2021a) that resamples inputs into a fixed amount of visual tokens. Finally, the self-attention layers of the language model are kept frozen during training while cross-attention layers are interleaved. A gating mechanism ensures that those new cross-attention layers do not interfere at model initialization, thereby improving stability and final performance.

Figure 3.74 shows the fundamental architecture of Flamingo. A pre-trained vision model as well as a pre-trained language model are frozen. Together they built the cornerstones of the model. The vision model is pre-trained using a contrastive text-image approach. Its role is to extract features such as colour, shape, nature and the position of objects - typical semantic spatial features that one would use in querying. The language model is an existing pre-trained language model. On top of those frozen parts, the authors add a perceiver-resampler and gated cross-attention layers as learnable architectures. The perceiver-resampler turns the outputs of the vision model into a fix set of visual tokens. Those visual tokens are then used to create cross-attention layers which are interleaved into the frozen language model. As a result, Flamingo

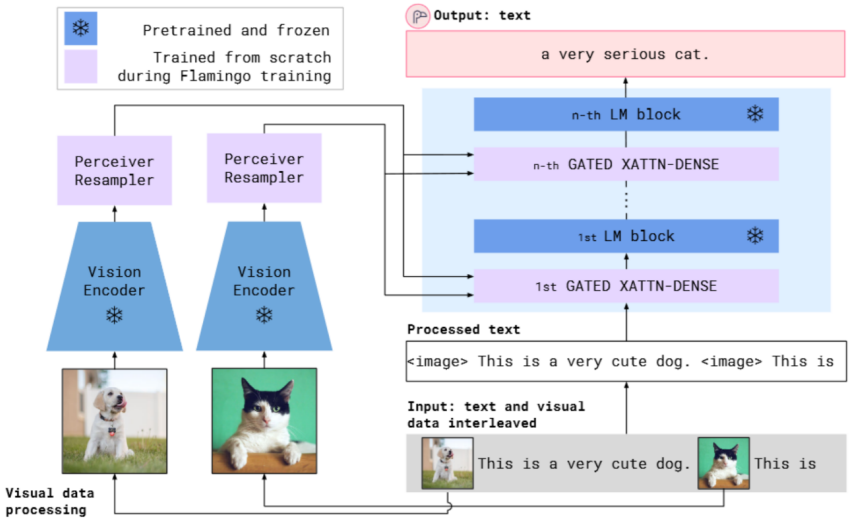


FIGURE 3.74: Alayrac et al. (2022): Flamingo Model Structure

can model the likelihood of some text y interleaved with a sequence of images or videos x as

$$p(y|x) = \prod_{l=1}^{L} p(y_l|y_{<l}, x_{<l}).$$

Here, y_l denotes the language token associated with the input text and $(y, x)_{<l}$ is the set of preceding tokens. Parameterized by the model is p. As shown in the initial figure, one conditions the Flamingo model simply by giving it an alternating image and text sequence. This is because the attention mechanism only allows text tokens to attend to the previous image, which turned out to work better than alternative schemes. In particular, this means that the model can generalize to an arbitrary amount of images regardless of the amount of images used in training.

The training was based on multiple different datasets. The most important one is a Multimodal MassiveWeb datatset (M3M). To generate it, the authors collect images and text from HTML of approximately 43 million webpages. In the process, the position of images relative to the surrounding text is also identified. Image tags (image) added in plain text signal the original location of images to the model. In addition to that, end of chunk (EoC) tokens before images separate image-text sequences. The embedding of that token is added to the vocabulary of the language model with a random initialization that can later be learnt. Then, it is possible to infer the length of an image-text sequence, which is another piece of derived information. To give an idea of the

scope of the dataset, M3M contains about 182GB of text as well as roughly 185 million images. The authors pay special attention not to include traditional task-specific datasets curated particularly for machine learning purposes to guarantee the generality of their modelling approach. As a second important dataset, aligned image text pairs are used. In particular, the ALIGN dataset (Jia et al., 2021b). The dataset is further augmented with Long Text and Image Pairs (LTIP) as well as Video and Text Pairs (VTP). The later datasets contain more descriptive captions than ALIGN. Together the process ensures that the available training datasets are sufficiently large and heterogeneous - two key properties necessary to hopefully achieve good few shot performance.

The training objective is to minimize the weighted sum of dataset specific expected negative log likelihood.

$$\sum_{m=1}^{M} \lambda_m \cdot \mathbb{E}_{(x,y)\sim\mathcal{D}_m}\left[-\sum_{\ell=1}^{L} \log p(y_\ell | y_{<\ell}, x_{\leq\ell})\right]$$

Each dataset is weighted with a scalar λ as datasets can be of different quality or feature different properties. Hence, it might be preferable to pay different attention to different datasets. According to the authors, tuning these weights was essential to overall performance. In practice, the optimization works as follows: a sample batch with visual language sequences from each dataset is used to compute the gradient of the loss in accordance to the weight of the dataset. Importantly, the authors find that it is beneficial to accumulate the gradients of all datasets before triggering an updating process. Naturally, the actual datasets used to train the models are extremely crucial, as well. In their ablation studies, the authors find that removing their web-scraped multimodal dataset from the training pool drops model performance as measured across all selected tasks from a score of 68.4 to 46.9. Removing the dataset containing aligned captions and images drops performance to a score of 56.5 and not accumulating gradients before the updating process decreases performance to 59.7.

Taking a closer look at the model architecture, the two key structures are the perceiver resampler and the attention layers. Figure 3.75 shows the architecture of the perceiver (Jaegle et al., 2021a). Before the input data reaches the perceiver, it is processed by the vision encoder - a Normalizer-Free ResNet which is trained with a constrastive loss similar to the well known Clip model (Radford et al., 2021a) and yields a good trade-off between performance and efficiency. The output of the vision encoder is a 2D grid which is than flatted before being fed into the perceiver that connects the vision encoder with the frozen language model. The resampling performed by the perceiver-resampler is crucial to reduce the complexity of vision-text cross-attention in the next step. This is particularly notable for video inputs. Inside the perceiver, a set of learned latent queries cross attend to the flattened vision encoder output. The number of outputs generated by the perceiver is equal to the number

of learned latent queries. A change the authors make compared to previous work is to concatenate the keys and values from the latent queries with the keys and values from the flattened features. The ablation studies show that a medium sized perceiver architecture works best with improvements around two to three score points. Furthermore, a too large architecture can lead to unstable trainings in conjunction with large frozen language model. The authors also test the perceiver against a transformer or MLP, which showed the perceiver to improve performance scores by around three to five points.

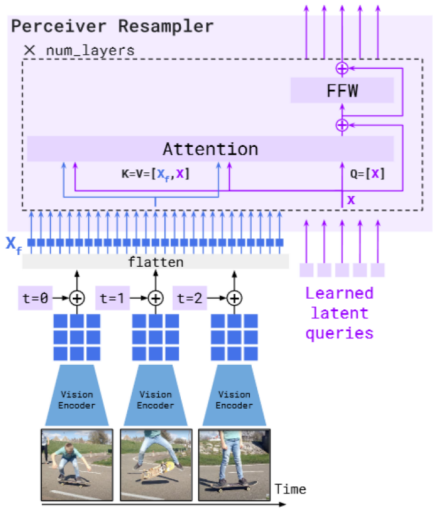


FIGURE 3.75: Alayrac et al. (2022): Flamingo Perceiver-Resampler

The main component that turns the frozen large language model into a functioning visual language model are the cross-attention layers depicted in figure 3.76. The number of layers added controls the number of free parameters and hence the complexity and expressiveness of the model. Keys and values of those layers are obtained from the visual features output by the perceiver while using language queries. Specifically, gated cross-attention dense layers are used. The layers are dense because the cross-attention layer is followed by a feed forward layer. They are gated because a $\tanh(\alpha)$ gating mechanism is applied between layers. The gating mechanism ensures that the frozen large language model remains stable at initialization through introducing a learnable scalar parameter α initialized at 0. Without that initialization, training instabilities can occur. A key question is how many cross-attention layers should be used. For the small Flamingo model with 3B parameters, the ablation studies show that adding cross-attention between every self-attention layer of the frozen model yields the best results. However, adding further cross-attention layers does notably scale the parameter count of the model. A clear performance trade-off exists. After making hardware considerations, the authors settled for

adding $\tanh(\alpha)$ gated cross-attention layers every 7th layer in the frozen large language model. The practical implementation of those attention layers works as follows: recall that the authors found that attending only to the nearest image improves performance by approximately 8 score points. To achieve this, while all text tokens attend to all visual tokens, a masking strategy is applied which ensures that in effect, language tokens only see a specific amount of visual tokens. Note however, that while the model can, unless specified otherwise, only attend to one image at a time, there is still a causal dependency to all previous images through the self-attention in the text-decoder.

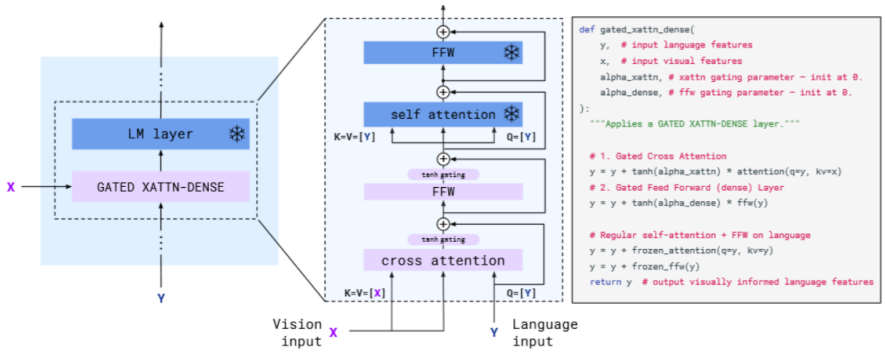


Figure 5 | **GATED XATTN-DENSE layers.** We insert new cross-attention layers, whose keys and values are obtained from the vision features while using language queries, followed by dense feed forward layers in between existing pretrained and frozen LM layers in order to condition the LM on visual inputs. These layers are *gated* so that the LM is kept intact at initialization for improved stability and performance.

FIGURE 3.76: Alayrac et al. (2022): Flamingo Gated Cross-Attention

To evaluate the model, the authors chose 18 different vison-language benchmarks including video benchmarks as shown in 3.77.

Note that seven benchmarks are used to validate design choices of the architecture. They are part of the development (dev) set. As those datasets could potentially report biased performance results, the remaining eleven datasets are solely used to estimate unbiased performance scores. Unfortunately, unbiased estimation in few-shot learning is not ubiquitous. Since hyperpameter tuning requires more prompts, it is easy to forget about them when counting how many shots in effect have been used, which can in turn lead to overestimation of performance (Perez et al., 2021b). As far as the Flamingo model is concerned, the authors take great care to evaluate it in a true few-shot fashion as coined by Perez et al. (2021b). Furthermore, most of the tasks require a generative answer (gen) which encompasses open-ended, more interesting tasks.

The results portayed in figure 3.78 show that Flamingo does not only outperform all previous few shot models, but also the general state-of-the art on six tasks. It is also, not too surprisingly, evident that more shots and larger models lead to better performance. The in-context learning works analogous

	Dataset	DEV	Gen.	Custom prompt	Task description	Eval set	Metric
Image	ImageNet-1k [103]	✓			Object classification	Val	Top-1 acc.
	MS-COCO [18]	✓	✓		Scene description	Test	CIDEr
	VQAv2 [3]	✓	✓		Scene understanding QA	Test-dev	VQA acc. [3]
	OKVQA [75]	✓	✓		External knowledge QA	Val	VQA acc. [3]
	Flickr30k [149]		✓		Scene description	Test (Karpathy)	CIDEr
	VizWiz [40]		✓		Scene understanding QA	Test-dev	VQA acc. [3]
	TextVQA [108]		✓		Text reading QA	Val	VQA acc. [3]
	VisDial [22]				Visual Dialogue	Val	NDCG
	HatefulMemes [60]			✓	Meme classification	Seen Test	ROC AUC
Video	Kinetics700 2020 [110]	✓			Action classification	Val	Top-1/5 avg
	VATEX [132]	✓	✓		Event description	Test	CIDEr
	MSVDQA [140]	✓	✓		Event understanding QA	Test	Top-1 acc.
	YouCook2 [161]		✓		Event description	Val	CIDEr
	MSRVTTQA [140]		✓		Event understanding QA	Test	Top-1 acc.
	iVQA [145]		✓		Event understanding QA	Test	iVQA acc. [145]
	RareAct [81]			✓	Composite action retrieval	Test	mWAP
	NextQA [139]		✓		Temporal/Causal QA	Test	WUPS
	STAR [138]				Multiple-choice QA	Test	Top-1 acc.

FIGURE 3.77: Alayrac et al. (2022): Flamingo Datasets (Table2, p.19)

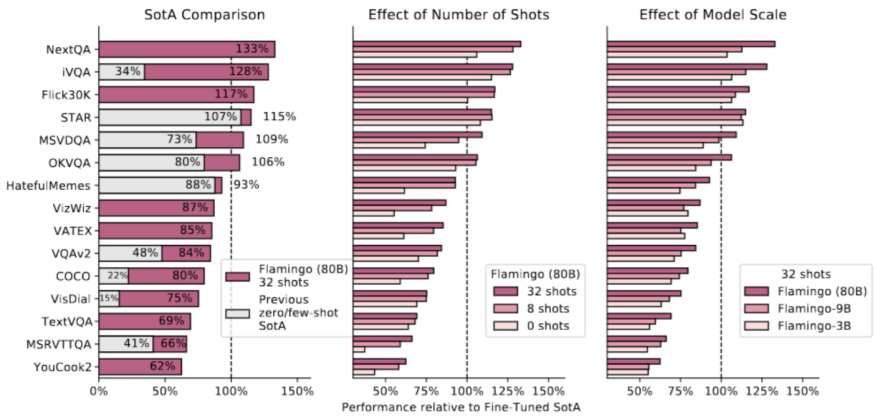


FIGURE 3.78: Alayrac et al. (2022): Flamingo Results without Fine-Tuning

to GPT-3 (Brown et al., 2020). Given a set of supporting examples (image, text), where text is the expected response based on the supporting visual input, a multimodal prompt is built by concatenating the examples in random order and adding the selected query image for the prediction. Interestingly, rather than using in-context learning with prompts, the authors also explore fine-tuning the model on the tasks which achieved state-of-the-art performance with few-shot learning. Fine-tuning the model is very expensive and requires additional hyperparameter tuning, but substantially improves results even further.

One notable exception that the authors remark upon is the classification performance of the model. In that realm, contrastive models outperform

Method	VQAV2		COCO	VATEX	VizWiz		MSRVTTQA	VisDial		YouCook2	TextVQA		HatefulMemes
	test-dev	test-std	test	test	test-dev	test-std	test	valid	test-std	valid	valid	test-std	test seen
Flamingo - 32 shots	67.6	-	113.8	65.1	49.8	-	31.0	56.8	-	86.8	36.0	-	70.0
SimVLM [134]	80.0	80.3	**143.3**	-	-	-	-	-	-	-	-	-	-
OFA [129]	79.9	80.0	149.6	-	-	-	-	-	-	-	-	-	-
Florence [150]	80.2	80.4	-	-	-	-	-	-	-	-	-	-	-
Flamingo Fine-tuned	**82.0**	**82.1**	138.1	**84.2**	**65.7**	**65.4**	**47.4**	61.8	59.7	118.6	**57.1**	54.1	**86.6**
Restricted SotA†	80.2 [150]	80.4 [150]	**143.3** [134]	76.3 [165]	- -	- -	46.8 [57]	**75.2** [87]	**74.5** [87]	**138.7** [142]	54.7 [147]	**73.7** [92]	75.4 [60]
Unrestricted SotA	81.3 [143]	81.3 [143]	149.6 [129]	81.4 [165]	57.2 [70]	60.6 [70]	- -	- -	75.4 [133]	- -	- -	- -	84.6 [164]

Table 6 | **Comparison to SotA when fine-tuning *Flamingo*.** We fine-tune *Flamingo* on all nine tasks where *Flamingo* was SotA overall with few-shot learning. *Flamingo* sets a new SotA on five of these tasks sometimes even beating methods that resorts to known performance optimization tricks such as model ensembling (on VQAv2, VATEX, VizWiz and HatefulMemes). Best numbers among the restricted SotA are in **bold**. Best numbers overall are underlined. Restricted SotA†: only includes methods that use a single model (not ensembles) and do not directly optimise the test metric (no CIDEr optimisation).

FIGURE 3.79: Alayrac et al. (2022): Flamingo Results with Fine-Tuning

Flamingo. A working hypothesis is that training for text-image retrieval is particularly important on those tasks. Nevertheless, few-shot learning and open-ended generative capabilities provide great advantages over contrastive models both in terms of model training as well as the range of applications. Flamingo shows how to leverage a unifying structure between models. Communication between models is key to reducing training overload and to enable multitasking models.

3.5.4 Discussion

In the previous sections, we've analyzed three different models with a common goal, to work towards a more general model capable of succeeding on multiple tasks across multiple modalities, specifically image and text. Which lessons can be learned from that?

Context is key. Across all three models, it became apparent that **larger models have an edge**. A clever architecture can of course lead to great results - after all Flamingo outperformed many fined-tuned models, but ceteris paribus, larger models will deliver stronger performance. While Flamingo with 80B parameters, mostly made up by its large language model component, was far larger than data2vec or VilBert, it is also far from the largest model in existence, as the following chapter will show. More tasks, more generalizability also means larger models.

Larger models in turn require either incredible resources or clever designs to be trained - likely both. It is not without reason that the three models presented have been developed by Meta and Google. At the high end, the amount of people with access to the necessary resources is naturally limited and there is little reason to expect change. On top of resources constraints limiting access

to mostly a few private companies, resulting models are often also simply not publicly available to shield intellectual property from competitors. For example, while the codebase for the somewhat older VilBert model is publicly available, data2vec and Flamingo are not generally accessible.

At any rate, even for the companies with the greatest resources, **a performance-cost trade-off exists**. The question of how to cut down on required training time is essential. The general approach is to pre-train models and then fine-tune them on specific tasks. However, **few shot in-context learning** provides a resource friendly alternative although fine-tuning likely still leads to better absolute results. **Freezing models**, particularly large language models, **is a key idea** on top of pre-training. In some cases, it is paramount to avoid the loss of capabilities that can go along with retraining a model. This could already be seen when VilBerts dual-stream architecture outperformed a single-stream design, but becomes more notable in the Flamingo architecture, where retaining the full expressiveness of the large language model is key which prompted the authors to introduce a gating mechanism into the cross-attention layers to stabilize the training process. In general, model collapse is always a concern, in particular when working with latent representations such as data2vec. In essence, rather than building single models from scratch, reusing models and leveraging communication between models is a new, promising approach. In that regard, Socratic models (Zeng et al., 2022) also show that the knowledge stored in different models is symbiotic which they used for exciting tasks such as multimodal assisted dialogue with people or robot perception. Finally, **data matters**. Not only is the amount of data important, but also its composition. Heterogeneous data are important, but so is the optimization across datasets. The Flamingo model was specifically trained with a weighted loss across datasets and it was possible to quantify the performance contribution of each of them. Particularly in few shot learning settings, it is thereby important to be careful about unbiased performance estimation as Perez et al. (2021b) noted. Otherwise, it is easy to overestimate performance.

In any case, the quest towards more general, more unified models is far from completed. The common theme is to combine larger and larger models while employing resource friendly training regimes. For example Pathway models (Chowdhery et al., 2022), which will be further discussed in the upcoming chapter, use sparse activation which reduces the energy consumption to less than 10% of what would be expected from similar dense models.

4

Further Topics

Authors: Marco Moldovan, Rickmer Schulte, Philipp Koch

Supervisor: Rasmus Hvingelby

So far we have learned about multimodal models for text and 2D images. Text and images can be seen as merely snapshots of the sensory stimulus that we humans perceive constantly. If we view the research field of multimodal deep learning as a means to approach human-level capabilities of perceiving and processing real-world signals then we have to consider lots of other modalities in a trainable model other than textual representation of language or static images. Besides introducing further modalities that are frequently encountered in multi-modal deep learning, the following chapter will also aim to bridge the gap between the two fundamental sources of data, namely structured and unstructured data. Investigating modeling approaches from both classical statistics and more recent deep learning we will examine the strengths and weaknesses of those and will discover that a combination of both may be a promising path for future research. Going from multiple modalities to multiple tasks, the last section will then broaden our view of multi-modal deep learning by examining multi-purpose modals. Discussing cutting-edge research topics such as the newly proposed Pathways, we will discuss current achievements and limitations of the new modeling that might lead our way towards the ultimate goal of AGI in multi-modal deep learning.

4.1 Including Further Modalities

Author: Marco Moldovan

Supervisor: Rasmus Hvingelby

Over the course of the previous chapters, we have introduced the basics of computer vision (CV) and natural language processing (NLP), after that we have learned about several directions of how we can combine these two subfields in machine learning. In the most general sense, we have explored ways in which we can process more than just one modality with our machine learning models.

So far, we have established the basics of multimodal deep learning by examining the intersection of these two most well-understood subfields of deep learning. These fields provide us with easy-to-handle data as seen in the corresponding previous chapter as well as a plethora of established and thoroughly examined models.

In reality though, text and images can be seen as only discrete snapshots of our continuous highly multimodal world. While text and images serve as an important foundation for us to develop concepts and algorithms for multimodal learning, they only represent a small part of what we as humans can perceive. First and foremost, we perceive reality in a temporal direction too, for a machine this could mean receiving video as input instead of just still images (IV et al., 2021). In fact, as videos are some of the most abundant types of data, we will later see that self-supervised learning on raw video is one of the major subtasks of multimodal deep learning. Clearly our reality is not just a sequence of RGB images though: just like in most videos we experience sound and speech which we would also like our models to process. Furthermore, we have different senses that can perceive depth, temperature, smell, touch, and balance among others. We also have sensors that can detect these signals and translate them to a digital signal so it is reasonable to want to have a machine learning algorithm detect and understand the underlying structure of these sensory inputs as well.

Now it might be tentative to simply list all different types of signals that we have developed sensors for and give a few examples of a state of the art (SOTA) deep neural network for each that tops some arbitrary benchmark. Since we are talking about multimodal learning, we would also have to talk about how these different modalities can be combined, and what the current SOTA research is, on all of these permutations of modalities. Quickly we would see that this list would get extremely convoluted and that we would not see the end of it. Instead of basing our understanding simply on a list of modalities we need a different, more intuitive system that lets us understand the multimodal research landscape. In the first part of this chapter we will attempt to introduce such a taxonomy based on challenges rather than modalities (**?**).

If we consider multimodal deep learning as the task to learn models that can perceive our continuous reality just as precisely (if not more) than us humans (LeCun, 2022), we have to ask ourselves how we can generalize our learnings from image-text multimodal learning to more types of signals. We have to ask what constitutes a different type of signal for us versus for a machine. What types of representation spaces we can learn if we are faced with having to process different signal types (modalities) and what are the strategies to learn these representation spaces. Here we will see that in large we can have two ways of processing modalities together, where defining their togetherness during training and inference will play the central role. After formalizing the types of multimodal representation learning we will move on

and elaborate what the fundamental strategies are that allow is to learn these representation spaces. Then again, we can ask what we can practically do with these representation spaces: Here the notion of sampling and retrieving from our learnt representation spaces will play a major role. In fact we will see that almost all practical multimodal tasks can be generalized to what we call multimodal translation, where given a signal in one modality we want to return a semantically related signal in another modality.

The ideas that were just introduced are in fact what we consider to be the central challenges of multimodal learning, these challenges constitute the main pillars of our taxonomy of multimodal deep learning. Every problem in multimodal learning will have to solve at least one of these challenges. By viewing multimodal deep learning through these lens we can easily come across a new modality and understand immediately how to approach this problem without breaking our taxonomy.

After understanding these challenges the reader will hopefully take home a new way of thinking about how to solve and understand multimodal problems. Hopefully, when coming across a new research paper and tackling a new research project the reader will identify the challenges that the paper is trying to solve or which challenge requires solving for the research project and immediately know where to look.

Looking at the broader spectrum of the AI research landscape, as Yann LeCun has done in his recent paper (LeCun, 2022), we can see that multimodal perception through deep learning is one particularly important building block for creating autonomous agents capable of displaying reason.

After having thoroughly introduced these central multimodal learning challenges we will look at some of the current research trends of multimodal deep learning from the point of view of our challenge taxonomy. In order to solve these challenges a system must implement two major building blocks: a multimodal model architecture and a training paradigm. In this part of the chapter we will introduce examples for both and successively generalize these concepts. By introducing more and more universal and problem- as well as modality-agnostic systems from current research we will lead into a research project that we ourselves are undertaking to merge a general multimodal model with a problem-agnostic training paradigm which will form the conclusion of this chapter. Hopefully by then two major concepts have transpired: 1) Introduce models and training paradigms that are general enough as to give a conclusion to this chapter's very title: learning from any and including an arbitrary amount of further modalities in our learner and 2) sticking to the analogy of the human perceptive system and presenting models and training paradigms that can learn from any type of input signal just like we humans can. In the spirit of Yann LeCun's JEPA paper the perceptive aspect of artificial intelligence is only one aspect of the system. Looking at the broader spectrum of the AI research landscape – as Yann LeCun has done in his recent paper,

we can identify that multimodal perception through deep learning is one particularly important building block for creating autonomous agents capable of displaying reason. Other aspects such as reasoning and especially multi-tasking and scaling will be elaborated in [this] following chapter.

4.1.1 Taxonomy of Multimodal Challenges

In this part we will introduce a taxonomy based on challenges within multimodal learning (**?**).

4.1.1.1 Multimodal Representation Learning

At the core of most deep learning problems lies representation learning: learning an expressive vector space of distributed embedding vectors in which we can define a distance function that informs us about the semantic relatedness of two data points in this learnt vector space. For the sake of simplicity, we will assume that these vector spaces are learnt via deep neural networks trained with backpropagation. Normally we will have to apply some preprocessing to our raw data in order to transform it into a format that a neural network can read, usually in the form of a 2-dimensional matrix. As output the neural network will return some high-dimensional vector. But what if we are presented with more than one signal type (i.e., multimodal input)? How do we structure our input so that our models can sensibly learn from this multimodal input?

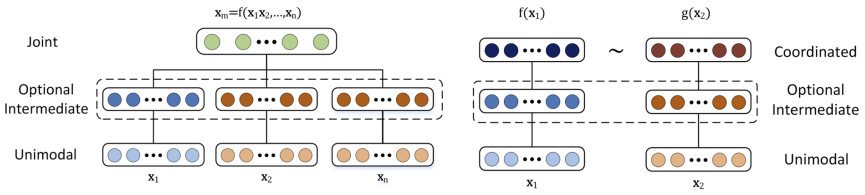


FIGURE 4.1: Joint and coordinated multimodal representations[@baltrušaitis2017multimodal].

In the introduction for this chapter, we briefly mentioned the togetherness of multimodal signals during training and inference (Bengio et al., 2013). By virtue of having more than one modality present as input into our learner – whether it be during training or inference – we want to relate these modalities somehow, this is the essence of multimodal learning. If we consider that our input signals from different modalities are somehow semantically related, we would like to leverage this relatedness across modalities and either have our learner share information between modalities and leverage this relatedness. Therefore cross-modal information has to come together at some point in our training and/or inference pipeline. How and when this happens is the

central question of multimodal representation learning which we describe in this subchapter.

First, we have to specify that what is meant by their togetherness during training and inference. Togetherness loosely means that unside our learner we "merge" the information of the modalities.

To make this more concrete: on one side we could think of concatenating the input from different modalities together to form one single input matrix. This joint input then represents a new entity that consists of multiple modalities but is treated as one coherent input. The model then learns one representation for the joint multimodal signal. On the other hand, we could think of the input always as strictly unimodal for one specific model. Each model would be trained on one modality and then the different modalities are brought together only in the loss function in such a way as to relate semantically similar inputs across modalities. To formalize what we just introduced, joint representation learning refers to projecting a concatenated multimodal input into one representation space while coordinated representation learning will learn different representation spaces for each modality and coordinate them such that we can sensibly align these representation spaces and apply a common distance function that can relate points across modalities to each other.

4.1.1.1.1 Joint Representations

Given for example a video that consist of a stream of RGB images and a stream of audio signals as a waveform we would like our model to learn a representation of this whole input video as how it appears "in the wild." Considering the entirety of the available input means that our model could leverage cross-modal information flow to learn better representations for our data: this means the model learns to relate elements from one modality to elements of the other. Of course, one could imagine concatenating all sorts of modalities together to feed into a model, such as audio and text, RGB image and depth maps, or text and semantic maps. The underlying assumption simply has to be that there is something to relate between the modalities – in other words there has to be a sensible semantic relationship between the modalities.

4.1.1.1.2 Coordinated Representation

When we are given data in multiple modalities, for learning coordinated representations, the underlying assumption will be that there exists some semantic relation between a signal in modality m and modality n. This relation can be equivalence – as in a video dataset where the audio at a given timestep t is directly intertwined with the sequence of RGB images at that timestep: they both are stand-ins for conceptually the same entity. The relation can also be a different function such as in the problem of cross-modal speech segment retrieval: here we want to return a relevant passage from an audio or speech file given a textual query. The text query is not the exact transcript of the

desired speech segment, but they do relate to each other semantically, for this our model would have to learn this complex relationship across modalities (**?**).

To do this we learn a class of models where each model will learn to project one modality into its own representation space. We then have to design a loss function in such a way as to transfer information from one representation to another: we essentially want to make semantically similar data points sit close together in representation space while having semantically dissimilar points sit far away from each other. Since each modality lives in its own representation space our loss function serves to align – or coordinate – these vector spaces as to fulfill this desired quality.

After having introduced what representation spaces we want to learn in the sections multimodal fusion and multimodal alignment we will elaborate further on exactly how we can learn joint and coordinate multimodal representation spaces respectively.

4.1.1.2 Multimodal Alignment

Alignment occurs when two or more modalities need to be synchronized, such as matching audio and video. It deals with the how rather than the what of learning coordinated representation spaces. Here, the goal is to learn separate representation spaces for each present modality, given that a dataset of corresponding data n-tuples exist. The embedding spaces are technically separate but through a carefully chosen learning strategy they are rotated and scaled such that their data points can be compared and queried across representation spaces. Currently the most common learning paradigm for alignment is contrastive learning. Contrastive learning was described extensively in a previous chapter, so in short: given a pair of semantically equivalent samples in different modalities we would want these data points to be as close as possible in embedding space while being far apart from semantically dissimilar samples(**?**).

4.1.1.3 Multimodal Fusion

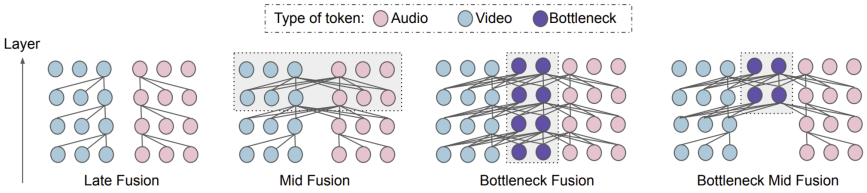


FIGURE 4.2: Different types of multimodal fusion[@baltrušaitis2017multimodal].

Analogous to alignment, multimodal fusion describes how joint representations

are learnt. Fusion describes the process of merging modalities inside the model, usually a concatenated and tokenized or patched multimodal input is fed into the model as a 2D matrix. The information from the separate modalities have to combine somehow inside the model to learn from one another to produce a more meaningful, semantically rich output. In the context of Transformer (Vaswani et al., 2017f) based models this usually means where the different inputs start attending to one another cross-modally. This can happen either early on in the model, somewhere in the middle, close to the output in the last layer(s) or based on a hybrid approach. These techniques are usually either based on heuristics, the researcher's intuition, biological plausibility, experimental evidence, or a combination of all [Nagrani et al. (2021)][@ DBLP:journals/jstsp/ZhangYHD20][@ shvetsova2021everything].

4.1.1.4 Multimodal Translation

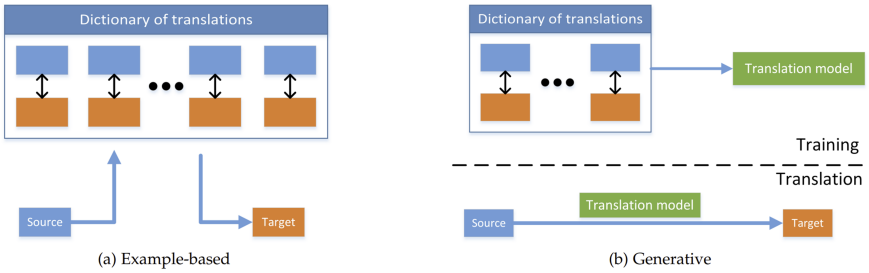


FIGURE 4.3: Different types of multimodal translation [@baltrušaitis2017multimodal].

In many practical multimodal use-cases we actually want to map from one modality to another: As previously mentioned we might want to return a relevant speech segment from an audio file given a text query, we want to return a depth map or a semantic map given an RGB image or we want to return a description of an image to read out for visually impaired people(Bachmann et al., 2022). In any way we are presented with a datapoint in one modality and want to translate it to a different modality. This another one of the main challenges of the multimodal deep learning landscape and it is what this subsection will be about (Sulubacak et al., 2020).

4.1.1.4.1 Retrieval

In order to perform cross-modal retrieval one essentially has to learn a mapping that maps items of one modality to items of another modality. Practically this means aligning separate unimodal representation spaces so that the neighborhood of one datapoint contains and equivalent datapoint of a different

modality when its representation space is queried at that point [Shvetsova et al. (2021)][@ DBLP:conf/eccv/Gabeur0AS20].

Currently cross-modal retrieval is almost exclusively learnt via contrastive learning which we described previously [Chen et al. (2020b)][@ oord2018representation][@ DBLP:conf/icml/ZbontarJMLD21].

4.1.1.4.2 Generation

We might also be presented with the case where we have a query in one modality but a corresponding datapoint in a different modality simply does not exist. In this case we can train generative multimodal translation models that learn to decode samples from a vector space into an output of a different modality. This requires us to learn models with a deep understanding of the structure of our data: when sampling datapoint from our cross-modal representation space and applying a decoder to produce the intended output we need to sample from a relatively smooth distribution (Zhang et al., 2020a). Since we are actually doing interpolation between known points of our data distribution, we want to produce sensible outputs from "in between" our original data. Learning this smooth distribution often requires careful regularization and appropriate evaluation poses another challenge(**?**).

With the hype around generative multimodal models created mostly by models such as Dall-E (Ramesh et al., 2021c) came a huge spike in research around this area [Saharia et al. (2022a)][@ wu2022nuwainfinity]. Currently lots of models generate photorealistic outputs through diffusion (Ho et al., 2020b), yet they still employ models such as a pretrained CLIP (Radford et al., 2021c) module as the backbone.

4.1.2 Current Research Trends: Generalized Self-Supervised Multimodal Perception

So far, we have understood the challenges we are faced with when trying to solve multimodal learning problems. We have understood that from a theoretical perspective we need to learn one or several semantic representation spaces and what the overarching constraints are for learning these vector spaces. Moreso, we have seen that given a coordinated representation space we can translate between modalities and decode our vector space into new data points. For joint representation spaces we can apply traditional downstream tasks such as classification or regression to better solve real world problems leveraging the interplay of all modalities at hand. Going forward we will explore the two major building blocks for realizing these challenges from a more practical perspective:

- Multimodal Architectures
- Multimodal Training Paradigms

A combination of carefully chosen model architecture and training scheme is necessary to solve the challenges we have described on a high level. Throughout the rest of this subchapter, we will look at more and more general concepts for each of these components. In this subchapter we will also connect back to one central principal that we have introduced earlier in this chapter: approaching human-level of multimodal perception. This means that we will follow one of the major lines of research within multimodal deep learning: building more general and problem-agnostic solutions. We pose the question: Why apply hyper-specific solutions when we can simplify and generalize our methods while retaining (or even improving) on experimental results. Towards the end of the chapter, we will also briefly introduce our own research in which we attempt to combine a modality agnostic model architecture with a generalized non-contrastive training paradigm for uni- and multi-modal self-supervised learning.

4.1.2.1 General Multimodal Architectures

First, we want to establish some desirable characteristics that our generalized multimodal model architectures should have:

- Input-Agnosticism: Whether our input consist of 1-dimensional sequences of audio waveforms or text or 3-dimensional inputs such as video we want out model to process all kinds of modalities equally with as little adjustments as possible.
- Multimodal Fusion: Ideally, we would also like to feed a concatenated input of several modalities into the model to learn joint representations.
- Preservation of Locality
- Respect Compositionality
- Flexible outputs: The model produces not only scalar or vector outputs but can ideally decode into any arbitrary output, thereby essentially having the capability for multimodal translation integrated.

We have not explicitly listed multimodal alignment as a desirable characteristic because the capability to perform alignment becomes trivial since we included the point about flexible outputs: to do alignment we need our model to output vectors that we can predict or regress over via a loss function. To illustrate the state of current research we will briefly introduce three multimodal model architecture that fulfill some, if not all of the above-mentioned criteria.

4.1.2.1.1 NÜWA

Initially conceived as a generative multimodal translation model here we are especially interested in the 3D-Nearby-Attention encoder-decoder stack at the center of NÜWA (Neural visUal World creAtion). We assume that our input, whether it be 1-dimensional sequences like text (or audio, although it was not used in the paper), 2-dimensional matrices like RGB images, sketches or semantic maps or 3-dimensional matrices like video (or possibly other

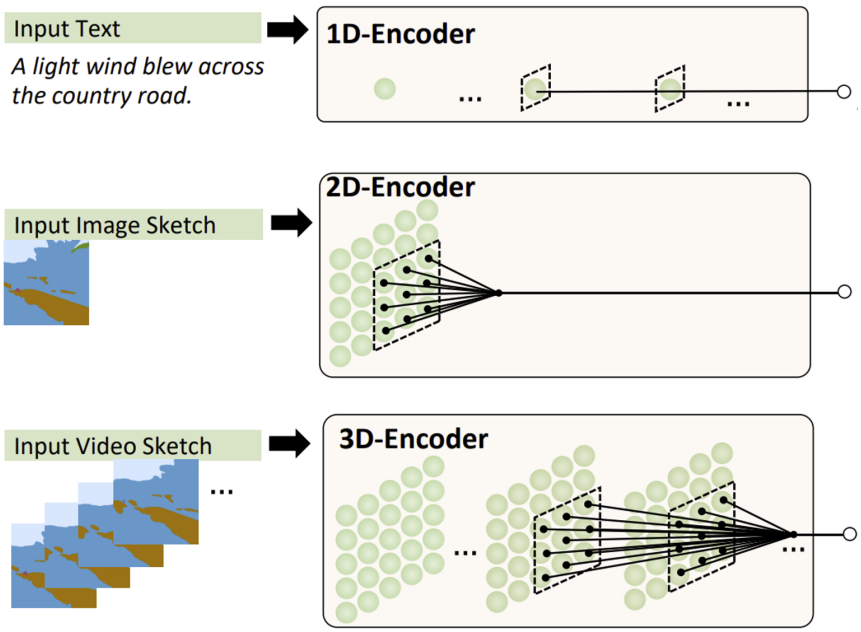


FIGURE 4.4: Data being represented in the NÜWA-imposed 3D format[@wu2021nwa].

modalities, e.g., depth maps) is already tokenized or divided into vectorized patches in case of 2D or 3D inputs. This is simply because we are using Transformer based encoder/decoder modules, therefore we need to reduce the input size beforehand. In the case of images or video this is done by a VQ-VAE. In principle any input is in the shape of a 3D matrix where one axis represents the temporal dimensions and the other axis representing the height and width. Clearly videos would fill out this input format in every direction, for still images the temporal dimension is simply one and for sequences such as text or audio they have height and width of one respectively and only stretch along the temporal dimension. Then for every token or patch a local neighborhood is defined amongst which the self-attention mechanism is applied. This saves on computational costs as for larger inputs global self-attention between all tokens or patches can become expensive. By imposing this 3D input format on all inputs, the model preserves the geometric and temporal structure of the original inputs, together with the locality respecting 3DNA mechanism the model introduces valuable and efficient inductive biases that make the model agnostic to input modality (if it is represented in the correct format), respects locality and allows for flexible outputs as it is intended to translate from any input to arbitrary outputs. Depending on how patching is performed for input

data one could imagine a setup where the 3D encoder-decoder could also implement a hierarchical structure which would also respect compositionality in data(Kahatapitiya and Ryoo, 2021), but this was not studied in this paper, although a similar idea was implemented in the follow-up paper [Wu et al. (2021)][@ wu2022nuwainfinity].

4.1.2.1.2 Perceiver IO

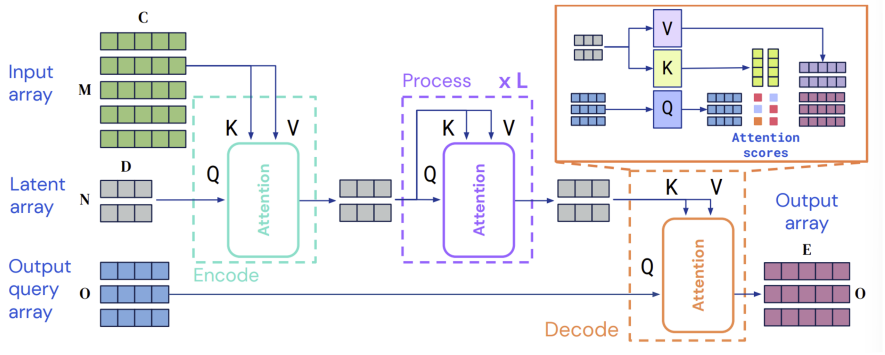


FIGURE 4.5: Perceiver encoder stack shows how the cross-attention mechanism transforms large inputs into smaller ones that can be processed by a vanilla Transformer encoder [@jaegle2021perceiver].

The Perceiver consists of a vanilla Transformer encoder block, with the follow-up paper, Perceiver IO, adding an analogous Transformer decoder in order to produce arbitrary multimodal outputs. The trick the Perceiver introduces in order to process nearly any sort of (concatenated) multimodal input is a specific cross-attention operation. Given a (very long) input in of size $M \times C$ a so-called latent array of size $N \times D$ is introduced, where $N \ll M$. With the input array acting as key and value and the latent array as the query a cross-attention block is applied between the two, this transforms the original input to a much smaller size, achieving higher than 300x compression. Perceiver IO is currently likely the most flexible model when it comes to processing and outputting arbitrary multimodal outputs, it also easily handles the learning of joint representation spaces at it can process very large input array of concatenated multimodal data such as long videos with audio or optical flow maps[Jaegle et al. (2021b)][@ jaegle2021perceiver].

4.1.2.1.3 Hierarchical Perceiver

With information about locality and compositionality being mostly lost in the Perceiver IO encoder this follow up paper imposes a hierarchical hourglass-like structure on the encoder-decoder stack. An input matrix with length M tokens is broken down into G groups, each M/G in size. For each group a separate

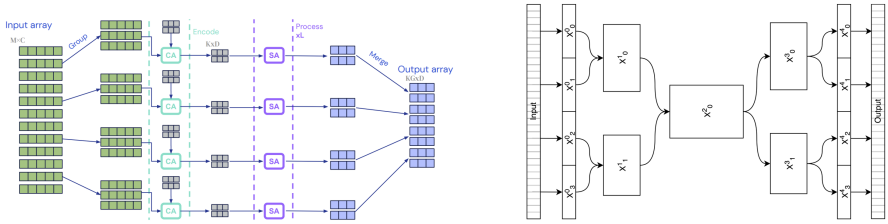


FIGURE 4.6: The hourglass structure of the Hierarchical Perceiver (HiP)[@carreira2022hierarchical].

latent array of size $K \times Z$ is initialized, and a cross-attention operation is applied between each group and its respective latent array, followed by a number of self-attention + MLP blocks. The set of output vectors of each group is then merged to form an intermediary matrix consisting of KG tokens. This intermediary matrix can be used as input to the next block, forming the hierarchical structure of the encoder. Besides embedding more of the locality and compositionality of data into the model, this architecture also improves upon computational costs on comparison to Perceiver IO (Carreira et al., 2022).

4.1.2.2 Multimodal Training Paradigms

Research of the past years has shown that in deep learning usually it is best to perform some sort of generalized task-agnostic pretraining routine. During self-supervised training that does not rely on any labeled data our model is trained to find underlying structures in the given unlabeled data. Given the right modeling and training scheme the model is able to approximate the true data distribution of a given dataset. This is extremely helpful as unlabeled data is extremely abundant so self-supervised learning lends itself as a way to infuse knowledge about the data into our model that it can leverage either directly for a downstream task (zero-shot) or helps as a running start during fine-tuning.

Since the conceptual part of pre-training was shown to have such an immense influence on downstream task performance, we will mainly focus on the self-supervised learning aspect of multimodal deep learning in this subchapter.

For self-supervised multimodal training paradigms, we can devise two major subcategories: those training paradigms that are agnostic to the input modality but operate only on a unimodal input and those that are both agnostic to input modalities but are truly multimodal.

4.1.2.2.1 Uni-Modal Modality-Agnostic Self-Supervised Learning

BYOL (Grill et al., 2020a) has introduced a paradigm shift for uni-modal self-supervised learning with its latent prediction mechanism. Its core idea is that the model to be trained is present in a student state and a teacher state where the teacher is a copy of the student with its weights updated by an exponentially moving average (EMA) of the student. Initially, BYOL was trained only on images: two augmentations would be applied to a base image and are then fed to the student and teacher network. The student would predict the latent states of last layers the teacher network via a simple regression loss. Data2vec extends this idea by generalizing it to other modalities: instead of applying specific augmentations to a base image a masking strategy is designed for each modality in order to augment the inputs, i.e., construct a semantically equivalent altered input. In the paper each modality has its own specific masking strategy and encoder backbone but in principle the paper showed that latent prediction SSL can be applied to other modalities such as text and audio just as well. Later we will introduce our own line of research where we try to generalize and simplify this even further and apply this concept to joint and coordinated representation problems.

Data2vec (Baevski et al., 2022) has already been extensively introduced in a previous chapter, because of that we would like to focus here on the importance of this relatively new line of SSL strategy that we call latent prediction SSL and why we think it is especially suitable for multimodal problems.

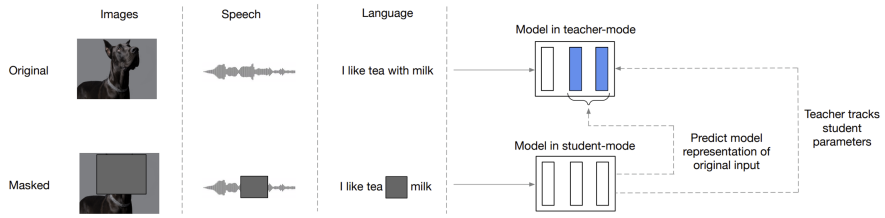


FIGURE 4.7: Illustration of data2vec[@baevski2022data2vec].

First off, how is latent prediction training different from other major SSL strategies like contrastive learning, masked auto-encoding (MAE) or masked token prediction? Whereas MAE and masked token prediction predict or decode into data space – meaning they need to employ a decoder or predict a data distribution – latent space prediction operates directly on the latent space. This means that the model has the predict the entire embedded context of an augmented or masked input, this forces the model to learn a better contextual embedding space and (hopefully) learn more meaningful semantic representations. Compared to contrastive approaches latent prediction methods do not require any hard negatives to contrast with. This alleviates us from the problem of producing hard negatives in the first place. Usually, they are

sampled in-batch at training time with nothing guaranteeing us they are even semantically dissimilar (or how much so) from the anchor. The loss function of latent prediction SSL is usually L1 or L2 regression loss which is easy and straight-forward and without the need to predict in data space or mine hard negatives we avoid many of the disadvantages of the other SSL training schemes while also improving upon contextuality of our embeddings by virtue of prediction in latent space (Baevski et al., 2022).

Since this training paradigm is generalizable to multimodal problems and avoids common points of failure of other major SSL strategies it is in line with the principle that we follow here, namely: Why solve a harder problem when we can simplify it and retain performance and generality?

4.1.2.2.2 Multimodal Self-Supervised Learning

As we have elaborated extensively, we are faced with learning either joint or coordinated representations in virtually any multimodal problem. Currently no learning framework or paradigm covers both joint and coordinated learning at the same time. Joint representations are usually learnt via contrastive methods whereas coordinated representations usually employ some variant of masked input prediction or denoising autoencoding.

As an example, for multimodal contrastive learning for joint representations we will first look at VATT (Akbari et al., 2021), short for Video-Audio-Text Transformer. In this paper the authors propose a simple framework for learning cross-modal embedding spaces across multiple (>= 2) modalities at once. They do so by introducing an extension to InfoNCE loss called Multiple-Instance-Learning-NCE (MIL-NCE). The model first linearly projects each modality into a feature vector and feeds it through a vanilla Transformer backbone. If the model is only used to contrast two modalities, then a normal InfoNCE loss is being used, for a video-audio-text triplet a semantically hierarchical coordinated space is learnt that enables us to compare video-audio and video-text by the cosine similarity. First a coordinated representation between the video and audio modality is constructed via the InfoNCE (Chen et al., 2020b) loss. Then a coordinated representation between the text modality and the now joint video-audio modality is also constructed similarly as shown in this figure. This hierarchy in these coordinated representations is motivated by the different levels of semantic granularity of the modalities, therefore this granularity is introduced into the training as an inductive bias. The aggregation of the several InfoNCE at different levels serves as the central loss function for this training strategy. It quickly becomes evident how this principle of learning (hierarchical) coordinated embeddings spaces can serve to learn between any n-tuples of different modalities if the respective n-tuples exist in a dataset.

MultiMAE (Bachmann et al., 2022) is a different kind of paper in which the authors learn a joint representation of a concatenated input consisting of RGB images, depth, and semantic maps. The input is partitioned into patches with

some of them randomly selected for masking. The flattened masked input is then fed into a multimodal ViT (Dosovitskiy et al., 2021) backbone. The authors then use different decoder blocks that act upon only the unimodal segments of the input. They hypothesize that the multimodal transformer can leverage cross-modal information in the input well enough as to embed multimodal semantic information in the output states. An additional global token that can access all input modalities is added for learning the joint representation. The task-specific decoders reconstruct their respective modality also by using one cross-attention module that can access information from the whole (multimodal) input. The aggregate reconstruction loss of all decoders serves as the model's loss function. This training strategy thereby produces a joint representation of an arbitrary ensemble of patched 2D modalities and can simultaneously learn to perform unimodal tasks as well.

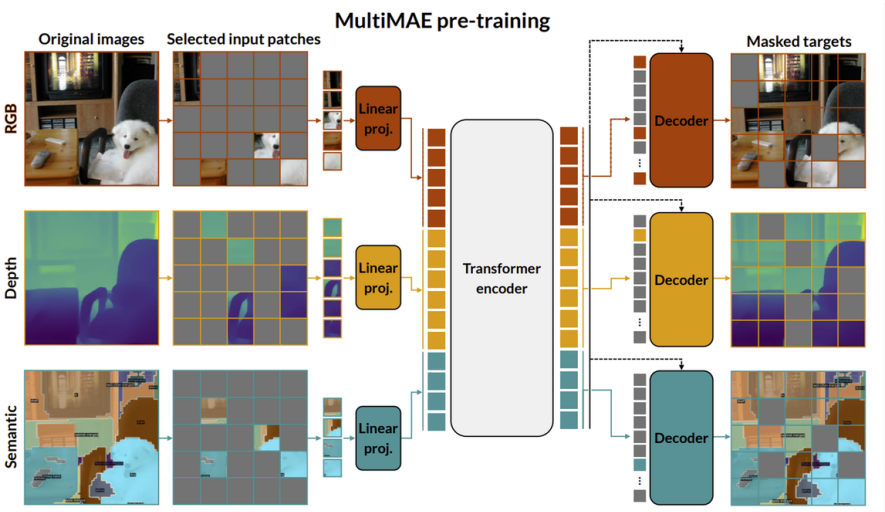


FIGURE 4.8: Masking and cross-modal prediction visualized for MultiMAE[@bachmann2022multimae].

So far, we have seen that for multimodal self-supervised learning we have a class of strategy that revolves around learning coordinated representations using contrastive learning. After having met contrastive multimodal models such as CLIP in earlier chapters we have shown that we can extend the same principles to include further modalities such as audio and video. In fact, if we are provided, with collections of multimodal data that pairs one modality to another, this principle can be applied to any set of given modalities.

Also, we have met training strategies that aim to generalize joint representation learning to multiple modalities. While the presented MultiMAE focuses on 2-dimensional modalities such as RGB, depth and semantic maps we could easily imagine applying the same principle to include other modalities as well

– given we can process our raw signals to represent them in the appropriate format a model can read.

We have omitted any specific learning strategies that pretrain specifically for translation tasks. For retrieval tasks it is evident that contrastive methods would offer zero-shot cross-modal retrieval capabilities. For generative tasks, the interested reader is invited to study the NÜWA paper whose 3D multimodal encoder we have introduced earlier: in it the authors leverage an identical 3D decoder to translate modalities one into another in a self-supervised manner. While the NÜWA 3D Attention encoder-decoder stack is not technically a multimodal model they do apply cross-modal cross-attention in to transfer semantic information from an encoded prompt to the decoder.

4.1.2.2.3 Personal Research: General Non-Contrastive Multimodal Representation Learning

So far, we have looked at unimodal and multimodal SSL as two separate categories. Research so far has not married the two concepts into a training paradigm that can learn both multimodal joint representations as well as cross-modal coordinated representations.

Let us consider a concatenated multimodal input signal. This concatenated input array would only really differ from a unimodal signal in that it already contains modality specific encodings added to the raw input – similar to those seen in the Perceiver. In fact, let us consider a multimodal input of the exact format seen in the Perceiver. In principle we could apply some masking strategy to this input array to mask out consecutive chunks of the input matrix and apply the same latent prediction training paradigm as seen in data2vec. We craft this masking strategy in such a way as to account for highly correlated nearby signals. If we were to randomly mask single rows of the input the task of predicting the mask input for very long inputs such as in videos or audio files becomes too trivial.

By representing all inputs, whether they are unimodal or multimodal in this unified format inspired by the Perceiver and applying a generic masking strategy we have essentially generalized data2vec to any arbitrary uni- or multimodal input. A Perceiver backbone model ensures that the handling and encoding of exceptionally large input arrays becomes efficient and effective.

Similarly let us consider a multimodal input for coordinated representations. Let us also assume that our model shares its weights across the separate modality-specific representation spaces (similar to VATT). Latent prediction training schemes such as BYOL and data2vec feed separate augmentations of the same input into the model which can be either in student or teacher (or online and offline) mode. The assumption is that both inputs should be roughly semantically equivalent so the model can learn to ignore the augmentations or masks to catch the essential structure within the data.

We pose the question: Are different modalities of the same thing also not just as semantically equivalent? Can we view different modalities simply as augmentations of one another and leverage the same training paradigm as in BYOL and data2vec, feeding one modality into the student model while feeding another into the teacher model? Would this learner be able to catch the essence of semantic equivalence of these two input signals? In our research project we try to answer these questions as well, our unified generalized multimodal learning framework is the first of its kind to be applicable to both joint as well as coordinated representations without any adjustments.

We propose this unified multimodal self-supervised learning framework as a novel and first-of-its-kind training paradigm that generalizes the unimodal self-supervised latent prediction training scheme inspired by BYOL and data2vec to an arbitrary number of input modalities for joint representation learning as well as cross-modal coordinated representation learning without the use of contrastive methods. Our method requires data to be presented in a generic format proposed by the Perceiver and requires just one single masking strategy.

This resolves the need for modality-specific masking strategies and models like in data2vec. For the cross-modal use-case we eliminate the need for hard negatives which are usually required for contrastive learning.

4.2 Structured + Unstructured Data

Author: Rickmer Schulte

Supervisor: Daniel Schalk

4.2.1 Intro

While the previous chapter has extended the range of modalities considered in multimodal deep learning beyond image and text data, the focus remained on other sorts of unstructured data. This has neglected the broad class of structured data, which has been the basis for research in pre-deep learning eras and which has given rise to many fundamental modeling approaches in statistics and classical machine learning. Hence, the following chapter aims to give an overview of both data sources and will outline the respective ways how these have been used for modeling purposes as well as more recent attempts to model them jointly.

Generally, structured and unstructured data substantially differ in certain aspects such as dimensionality and interpretability. This has led to various modeling approaches that are particularly designed for the special characteristics of the data types, respectively. As shown in previous chapters, deep learning

models such as neural networks are known to work well on unstructured data. This is due to their ability to extract latent representation and to learn complex dependencies from unstructured data sources to achieve state-of-the art performance on many classification and prediction tasks. By contrast, classical statistical models are mostly applied on tabular data due the advantage of interpretability inherent to these models, which is commonly of great interest in many research fields. However, as more and more data has become available to researchers today, they often do not only have one sort of data modality at hand but both structured and unstructured data at the same time. Discarding one or the other data modality makes it likely to miss out on valuable insights and potential performance improvements.

Therefore, in the following sections we will investigate different proposed methods to model both data types jointly and examine similarities and differences between those. Different fusion strategies to integrate both types of modalities into common deep learning architectures are analyzed and evaluated, thereby touching upon the concept of end-to-end learning and its advantages compared to separated multi-step procedures. The different methods will be explored in detail by referring to numerous examples from survival analysis, finance and economics. Finally, the chapter will conclude with a critical assessment of recent research for combining structured and unstructured data in multimodal DL, highlighting limitations and weaknesses of past research as well as giving an outlook on future developments in the field.

4.2.2 Taxonomy: Structured vs. Unstructured Data

In order to have a clear setup for the remaining chapter, we will start off with a brief taxonomy of data types that will be encountered. Structured data, normally stored in a tabular form, has been the main research object in classical scientific fields. Whenever there was unstructured data involved, this was normally transformed into structured data in an informed manner. Typically, doing so by applying expert knowledge or data reduction techniques such as PCA prior to further statistical analysis. However, DL has enabled unsupervised feature extraction from unstructured data and thus to feed it to the models directly. Classical examples of unstructured data are image, text, video, and audio data as shown in the figure below. Of these, image data in combination with tabular data is the most frequently encountered. Hence, this combination will be examined along various examples later in the chapter. While previously mentioned data types allow for a clear distinction, lines can become increasingly blurred. For example, the record of a few selected biomarkers or genes from patients would be regarded as structured data and normally be analyzed with classical statistical models. On the contrary, having the records of multiple thousand biomarkers or genes would rather be regarded as unstructured data and usually be analyzed using DL techniques. Thus, the distinction between structured and unstructured data does not only follow

along the line of dimensionality but also concerns the interpretability of single features within the data.

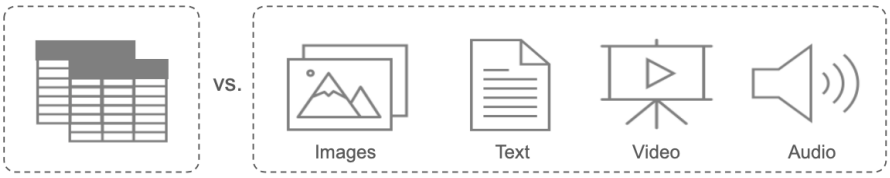


FIGURE 4.9: Structured vs. Unstructured Data

4.2.3 Fusion Strategies

After we have classified the different data types that we will be dealing with, we will now discuss different fusion strategies that are used to merge data modalities into a single model. While there are potentially many ways to fuse data modalities, a distinction between three different strategies, namely early, joint and late fusion has been made in the literature. Here we follow along the taxonomy laid out by Huang et al. (2020) with a few generalizations as those are sufficient in our context.

Early fusion refers to the procedure of merging data modalities into a common feature vector already at the input layer. The data that is being fused can be raw or preprocessed. The step of preprocessing usually involves dimensionality reduction to align dimensions of the input data. This can be done by either training a separate DNN (Deep Neural Network), using data driven transformations such as PCA or directly via expert knowledge.

Joint fusion offers the flexibility to merge the modalities at different depths of the model and thereby to learn latent feature representations from the input data (within the model) before fusing the different modalities into a common layer. Thus, the key difference to early fusion is that latent feature representation learning is not separated from the subsequent model. This allows backpropagation of the loss to guide the process of feature extraction from raw data. The process is also called end-to-end learning. Depending on the task, CNNs or LSTMs are usually utilized to learn latent feature representations. As depicted in the figure below, it is not required to learn lower dimensional feature representations for all modalities and is often only done for unstructured data. A further distinction between models can be made regarding their model head, which can be a FCNN (Fully Connected Neural Network) or a classical statistical model (linear, logistic, GAM). While the former can be desirable to capture possible interactions between modalities, the latter is still frequently used as it preserves interpretability.

Late fusion or sometimes also called decision level fusion is the procedure of fusing the predictions of multiple models that have been trained on each data

modality separately. The idea originates from ensemble classifiers, where each model is assumed to inform the final prediction separately. Outcomes from the models can be aggregated in various ways such as averaging or majority voting.

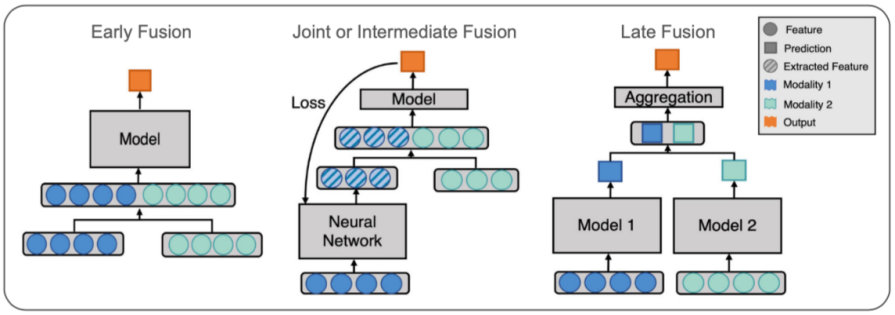


FIGURE 4.10: Data Modality Fusion Strategies (Adopted from Huang et al., 2020).

We will refer to numerous examples of both early and joint fusion in the following sections. While the former two are frequently applied and easily comparable, late fusion is less common and different in nature and thus not further investigated here. As a general note, for the sake of simplicity we will refer to the special kind of multimodal DL including both structured and unstructured data when we speak about multimodal DL in the rest of the chapter.

4.2.4 Applications

The following section will discuss various examples of this kind of multimodal DL by referring to different publications and their proposed methods. The publications originate from very different scientific fields, which is why methods are targeted for their respective use case. Hence, allowing the reader to follow along the development of methods as well as the progress in the field. Thereby, obtaining a good overview of current and potential areas of applications. As there are various publications related to this kind of multimodal DL, the investigation is narrowed down to publications which either introduce new methodical approaches or did pioneering work in their field by applying multimodal DL.

4.2.4.1 Multimodal DL in Survival

Especially in the field of survival analysis, many interesting ideas were proposed with regards to multimodal DL. While clinical patient data such as electronic health records (EHR) were traditionally used for modeling hazard functions in survival analysis, recent research has started to incorporate image data such as

CT scans and other modalities such as gene expression data in the modeling framework. Before examining these procedures in detail, we will briefly revisit the classical modeling setup of survival analysis by discussing the well-known Cox Proportional Hazard Model (CPH).

4.2.4.2 Traditional Survival Analysis (CPH Model)

Survival Analysis generally studies the time duration until a certain event occurs. While many methods have been developed to analyze the effect of certain variables on the survival time, the Cox Proportional Hazard Model (CPH) remains the most prominent one. The CPH model models the hazard rate which is the conditional probability of a certain event occurring in the next moment given that it has not so far:

$$h(t|x) = h_0(t) * e^{x\beta}$$

where $h_0(t)$ denotes the baseline hazard rate and β the linear effects of the covariates x on which the probability is conditioned on. The fundamental assumption underlying the traditional CPH is that covariates influence the hazard rate proportionally and multiplicatively. This stems from the fact that the effects in the so-called risk function $f(x) = x\beta$ are assumed to be linear. Although this has the advantage of being easily interpretable, it does limit the flexibility of the model and thus also the ability to capture the full dynamics at hand.

4.2.4.3 Multimodal DL Survival Analysis

Overcoming the limitations of the classical CPH model, Katzman et al. (2018) were among the first to incorporate neural networks into the CPH and thereby replacing the linear effect assumption. While their so-called DeepSurv model helped to capture interactions and non-linearities of covariates, it only allowed modeling of structured data. This gave rise to the model DeepConvSurv of Zhu et al. (2016), who apply CNNs to extract information from pathological images in order to predict risk of patients subsequently. They showed that learning features from images via CNNs in an end-to-end fashion outperforms methods that relied on hand-crafting features from these images. Building on the idea of DeepConvSurv, Yao et al. (2017) extended the model by adding further modalities. Besides pathological images, their proposed DeepCorrSurv model also includes molecular data of cancer patients. The name of the model stems from the fact that separate subnetworks are applied to each modality and that the correlation between the output of these modality specific subnetworks are maximized before fine-tuning the learned feature embedding to perform well on the survival task. The correlation maximization procedure aims to remove the discrepancy between modalities. It is argued that the procedure is beneficial in small sample settings as it may reduce the impact of noise inherent to a single modality that is unrelated to the survival prediction task.

The general idea is that the different modalities of multimodal data may contain both complementary information contributed by individual modalities as well as common information shared by all modalities. The idea was further explored by subsequent research. Tong et al. (2018) for example introduced the usage of auto encoders (AE) in this context by proposing models that extract the lower dimensional hidden features of the AE applied to each modality. While their first model trains AEs on each modality separately before concatenating the learned features (ConcatAE), their second model obtains cross-modality AEs that are trained to recover both modalities from each modality respectively (CrossAE). Here, the concept of complementary information of modalities informing survival prediction separately gives rise to the first model, whereas the concept of retrieving common information inherent across modalities gives rise to the latter. Although, theoretically both models could also handle classical tabular EHR data, they were only applied to multi-omics data such as gene expressions of breast cancer patients.

Similar to Tong et al. (2018), Cheerla and Gevaert (2019) also derive their model from the idea of common information that is shared by all modalities. Besides, having specialized subnetworks for each modality to learn latent feature embeddings, they also introduce a similarity loss that is added to the classical cox loss from the survival prediction. This similarity loss is applied to each subnetwork output and aims to learn modality invariant latent feature embeddings. This is desirable not only for noise reduction but also in cases of missing data. While previous research often applied their models only on subsets of the large cancer genome atlas program (TCGA), Cheerla and Gevaert (2019) analyze 20 different cancer types of the TCGA using four different data modalities. As expanding the scope of the study increases the problem of data missingness, they specifically target the problem by introducing a variation of regular dropout, which they refer to as multimodal dropout. Instead of dropping certain nodes, multimodal dropout drops entire modalities during training in order to make models less dependent on one single data source. This enables the model to better cope with missing data during inference time. Opposed to Tong et al. (2018), the model is trained in an end-to-end manner and thus allows latent feature learning to be guided by the survival prediction loss. More impressive than their overall prediction performances are the results of T-SNE-mappings that are obtained from the learned latent feature embeddings. One sample mapping is displayed in the figure below, which nicely shows the clustering of patients with regards to cancer types. This is particularly interesting regarding the fact that the model was not trained on this variable. Besides being useful for accurate survival prediction, such feature mappings can directly be used for patient profiling and are thus pointed out as a contribution to the research on their own.

Vale-Silva and Rohr (2021) extend the previous work by enlarging the scope of study, analyzing up to six different data modalities and 33 cancer types of the TCGA dataset. Their so-called MultiSurv model obtains a straightforward

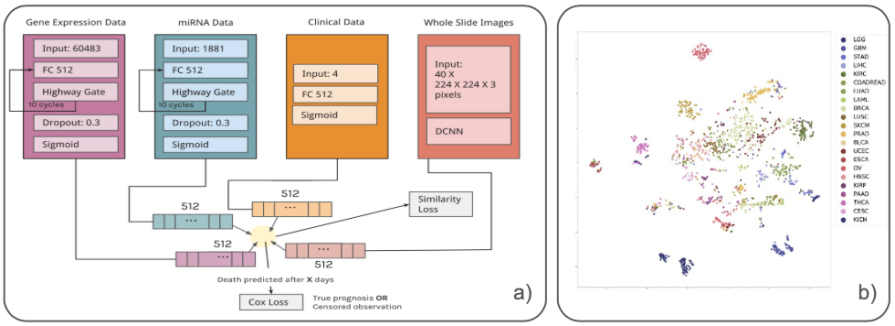


FIGURE 4.11: a) Architecture with Similarity Loss b) T-SNE-Mapped Representations of Latent Features (Colored by Cancer Type) (Cheerla and Gevaert, 2019).

architecture, applying separate subnetworks to each modality and a subsequent FCNN (model head) to yield the final survival prediction. Testing their modular model on different combinations of the six data modalities, they find the best model performance for the combination of structured clinical and mRNA data. Interestingly, including further modalities lead to slight performance reductions. Conducting some benchmarking, they provide evidence for their best performing model (structured clinical + mRNA) to outperform all single modality models. However, it is worthwhile mentioning that their largest model, including all six modalities, is not able to beat the classical CPH model, which is based on structured clinical data only. While this already may raise concerns about the usefulness of including so many modalities in the study, high variability of model performance between the 33 cancer types is also found by the authors and may indicate a serious data issue. The finding may seem less surprising, considering the fact that tissue appearances can differ vastly between cancer types. This is particularly problematic as for some of these cancer types only very few samples were present in the training data. For some there were only about 20 observations in the training data. Although state-of-the-art performance is claimed by the authors, the previously mentioned aspects do raise concerns about the robustness of their results. Besides, facing serious data quantity issues for some cancer types, results could simply be driven by the setup of their analysis by testing the model repeatedly on different combinations of data modalities. Thereby increasing the chances to achieve better results at least for some combinations of data modalities. Moreover, the study nicely showcases that the most relevant information can often be retrieved from classical structured clinical data and that including further modalities can by contrast even distort model training when sample sizes are low compared to the variability within the data. While these concerns could certainly have been raised for the other studies as well, they simply become

more apparent in Vale-Silva and Rohr (2021) due their comprehensive and transparent analysis.

In the last part of this section we will refer to a different set of survival models by introducing the concept of Wide & Deep NN. The idea for Wide & Deep NN was first introduced by Cheng et al. (2016), who proposed to not only feed data inputs to either a linear or FCNN model part, but both at the same time. Applying it in the context of Recommender Systems, the initial assumption was that models need to be able to memorize as well as generalize for prediction tasks and that these aspects could be handled by the linear and FCNN part, respectively.

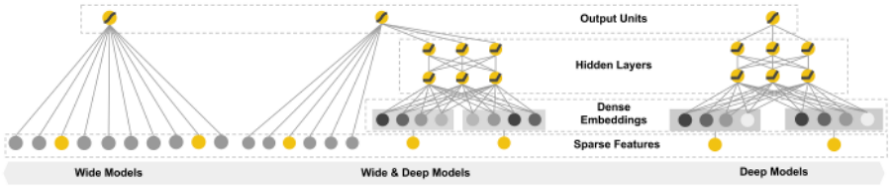


FIGURE 4.12: Illustration of Wide & Deep Neural Networks (Cheng et al., 2016).

The idea of Wide & Deep NN is applied in the context of multimodal DL survival by Pölsterl et al. (2019) and Kopper et al. (2022). Similar to previous studies Pölsterl et al. (2019) make use of the CPH model and integrate Wide & Deep NN in these. By contrast, Kopper et al. (2022) integrate them in a different set of survival models, namely the piecewise exponential additive mixed model (PAMM). The general purpose of this model class is not only to overcome the linearity but also the proportionality constraint in the classical CPH. By dropping the proportionality assumption, these models yield piecewise constant hazard rates for predetermined time intervals. Although the two studies differ in their model setup, both studies leverage structured as well as visual data and additionally make use of a linear model head. The latter is particularly interesting as it is this additive structure in the last layer of the models which preserves interpretability. Thus, they obtain models that not only have the flexibility for accurate predictions themselves but which are also able to recover the contributions of single variables to these predictions.

Although, Wide & Deep NN are advantageous due to their flexibility and interpretability, special care needs to be taken regarding a possible feature overlap between the linear and NN part as it can lead to an identifiability problem. This can be illustrated by considering the case that a certain feature x is fed to the linear as well as the FCNN model part. Because of the Universal Approximation Theorem for Neural Networks, it is known that the FCNN part could potentially model any arbitrary relation between the dependent and independent variable ($d(x)$). However, this is what raises the identifiability issue as the coefficients (β) of the linear part could theoretically be altered

arbitrarily ($\widetilde{\beta}$) without changing the overall prediction when the weights of the NN ($\widetilde{d}(x)$) are adjusted accordingly.

$$x\beta + d(x) = x\widetilde{\beta} + d(x) + f(x) = x\widetilde{\beta} + \widetilde{d}(x)$$

Generally, there are two ways to deal with this identifiability problem. The first possibility would be to apply a two-stage procedure by first estimating only the linear effects and then applying the DL model part only on the obtained residuals. An alternative way would be to incorporate orthogonalization within the model, thereby performing the procedure in one step and allowing for efficient end-to-end training. The latter was proposed by Rügamer et al. (2020) and utilized in the DeepPAMM model by Kopper et al. (2022). The next section will go into more detail about the two possibilities to solve the described identifiability issue and proceed by discussing further applications of multimodal DL in other scientific fields.

4.2.4.4 Multimodal DL in Other Scientific Fields

After having seen multiple applications of multimodal DL in survival analysis which predominantly occurs in the biomedical context, we will now extend the scope of the chapter by discussing further applications of multimodal DL related to the field of economics and finance. While structured data has traditionally been the main studied data source in these fields, recent research has not only focused on combining both structured and unstructured data, but also on ways to replace costly collected and sometimes scarcely available structured data with freely available and up-to-date unstructured data sources using remote sensing data. Before examining these approaches, we will first go into more detail about the model proposed by Rügamer et al. (2020), which not only introduced a new model class in the context of multimodal DL but also offered a method to efficiently solve the above mentioned identifiability problem.

As previous research exclusively focused on mean prediction, uncertainty quantification has often received less attention. Rügamer et al. (2020) approach this by extending structured additive distributional regression (SADR) to the DL context. Instead of learning a single parameter e.g. the mean, SADR provides the flexibility to directly learn multiple distributional parameters and thereby natively includes uncertainty quantification. It is nevertheless possible to only model the mean of the distribution, which is why SADR can be regarded as a generalization of classical mean prediction. Rügamer et al. (2020) now extend this model class by introducing a framework that can model these distributional parameters as a function of covariates via a linear, generalized additive (GAM) or NN model. All distributional parameters are resembled in a final distributional layer (output layer). An illustration of their so-called Semi-Structured Deep Distributional Regression (SSDDR) is given in the figure below.

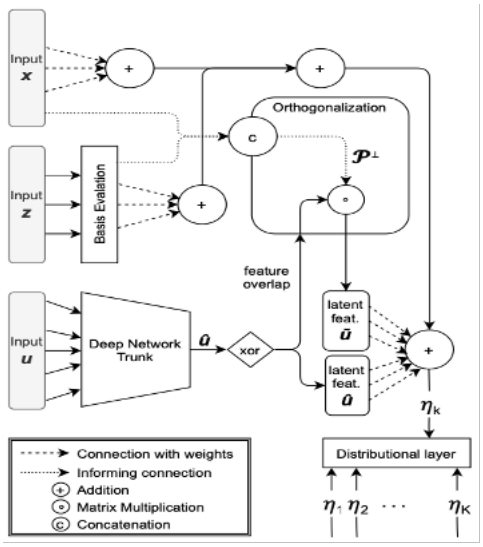


FIGURE 4.13: Architecture of SSDDR (X+Z (Struct.) and U (Unstruct.) Data) (Rügamer et al., 2020).

If the mean is now modeled by both a linear and DNN part and the same feature inputs are fed to both model parts, we are in the setting of Wide & Deep NN. As illustrated above, such feature overlaps give rise to an identifiability issue. The key idea to mitigate this problem from Rügamer et al. (2020) was to integrate an orthogonalization cell in the model, that orthogonalizes the latent features of the deep network part with respect to the coefficients of the linear and GAM part if feature overlaps are present. More precise, in case $\boldsymbol{X}$ contains the inputs, that are part of the feature overlap, the projection matrix $\boldsymbol{\mathcal{P}}^{\perp}$ projects into the respective orthogonal complement of the linear projection which is on the column space spanned by $\boldsymbol{X}$. This allows backpropagation of the loss through the orthogonalization cell and therefore enables end-to-end learning. As the linear and GAM effect channels are directly connected to the distributional layer, the orthogonalization cell is therefore able to preserve the interpretability of the model.

Another way of orthogonalizing feature representations is by applying a two-stage procedure as described above. Law et al. (2019) utilize this procedure to make their latent feature representations retrieved from unstructured data orthogonal to their linear effect estimates from structured data. More specifically, they try to accurately predict house prices in London using multimodal DL on street and aerial view images as well as tabular housing attributes. Applying the two-stage procedure they aim at learning latent feature representations from the image data which only incorporate features that are orthogonal to the housing attributes. Thereby, they limit the chances of confounding in

order to obtain interpretable housing attribute effects. Conducting a series of experiments, they find that including image data next to the tabular housing data does improve the prediction performance over single modality models albeit structured data remains the most relevant single data source. As a next step, they test their models with different model heads as depicted in the figure below to explore their respective potentials. Although fully nonlinear models with a DNN as model head generally offer larger modeling flexibility, as they can incorporate interactions, they achieved only slight performance gains over the semi-interpretable models with additive linear model heads. This is particularly interesting as the latter additionally preserve the often desired interpretability of effects. As the semi-interpretable models perform reasonably well, the authors argue that it is indeed possible to obtain interpretable models without losing too much on the performance side.

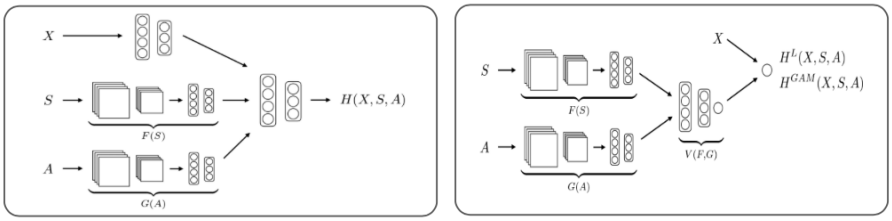


FIGURE 4.14: Fully Nonlinear and Semi-Interpretable Models (X (Struct.) and S+A (Unstruct.) Data) (Law et al., 2019).

In the last part of this section, we will allude to several other promising approaches that did pioneering work related to multimodal DL. While most of them use unstructured data sources such as remote sensing data, some do not specifically include structured data. They are still covered in this chapter to give the reader a broad overview of current research in the field. Moreover, structured data could easily be added to each of these models, but often studies intentionally avoid the use of structured data sources as they are sometimes scarcely available due to the cost of data collection. Besides availability, structured data such as household surveys is often irregularly collected and differs vastly between countries, making large scale studies impossible. Therefore, different studies have tried to provide alternatives to classical surveys by applying DL methods on freely available unstructured data sources. While Jean et al. (2016) use night and daylight satellite images to predict poverty in several African countries, Gebru et al. (2017) use Google Street View images to estimate socioeconomic attributes in the US. Both deploy the classical DL framework such as CNNs to retrieve relevant information from image data for the prediction task. Achieving reasonable prediction results while keeping analysis costs at low levels, both studies outline the potential of their proposed methods as being serious alternatives to current survey based analysis.

Other studies such as You et al. (2017) and Sirko et al. (2021) proposed DL frameworks for satellite imagery in contexts where labelled data is normally scarce. While You et al. (2017) use Deep Gaussian Processes to predict corn yield in the US, Sirko et al. (2021) apply CNNs to detect and map about 516 million buildings across multiple African countries (around 64% of the African continent). Besides being of great importance for applications such as commodity price predictions or financial aid distribution, the results of the two studies could easily be combined with other structured data sources and thereby could constitute a form of multimodal DL with high potential.

4.2.5 Conclusion and Outlook

In the previous sections we have come across various methods of multimodal DL that can deal with both structured and unstructured data. While these often differed substantially in their approach, all of them had in common that they tried to overcome limitations of classical modeling approaches. Examining several of them in detail, we have seen applications of different fusion strategies of data modalities and thereby touched upon related concepts such as end-to-end learning. The issue of interpretability was raised along several examples by discussing the advantages of different model heads as well as ways to solve identifiability problems using orthogonalization techniques.

It was indeed shown that it is possible to obtain interpretable models that are still capable of achieving high prediction performances. Another finding of past research was that end-to-end learning frequently showed to be superior compared to methods which learn feature representation via independent models or simply retrieve information via expert knowledge. Furthermore, research that actually conducted a comparison between their proposed multimodal DL and single modality models, almost always found their proposed multimodal model to outperform all models which were based on single modalities only. Nevertheless, within the class of single modality models, those using only structured data usually performed best. This leads to the conclusion that structured data often incorporates the most relevant information for most prediction tasks. By contrast, unstructured data sources may be able to add supplementary information and thereby partially improve performances.

While there certainly has been a lot of progress in the field of multimodal DL, conducted analyses still have their limitations which is why results need to be considered with care. Although most research finds their proposed multimodal DL models to achieve excellent performances, not all of them conduct benchmarking with regard to single modality models. Thereby, they limit the possibility to properly evaluate actual improvements over classical modeling approaches. Another aspect that may raise concerns regarding the reliability of results is that multimodal DL models such as most deep learning models have multiple hyperparameters. Together with the flexibility of choosing from a wide variety of data modalites, it opens up the possibility to tune the multimodal

models in various ways. Thereby making it possible that actual performance improvements may only be existent for certain configurations of the model as well as combinations of data modalities. This problem is likely to be empathized for studies using only small datasets. Small datasets are especially common in the biomedical context where image data of certain diseases is normally scarce. On top of the previously mentioned aspects, publication bias may be a large problem in the field as multimodal DL models that do not show improvements over single modality or other existing benchmark models, are likely to not be published.

Although there might be concerns regarding the robustness and reliability of some results, past research has surely shown promising achievements that could be extended by future research. While small sample sizes especially for unstructured data such as clinical images were outlined as a great limitation of past research, more of such data will certainly become available in the future. As deep learning methods usually require large amounts of training data to uncover their full potential, the field will probably see further improvements once sufficiently large datasets are available. Hence, including only an increasing number of modalities with limited samples in the models will likely be insufficient. Instead, the most promising approach seems to be incorporating sufficiently large data amounts of certain unstructured and structured data modalities that contain relevant information for the problem at hand.

4.3 Multipurpose Models

Author: Philipp Koch

Supervisor: Rasmus Hvingelby

In this chapter, we will broaden the focus to include multitask learning additionally to multimodal learning. We will call this approach multipurpose models. Many multipurpose models have been introduced in recent years (Kaiser et al. (2017), Hu and Singh (2021b), Wang et al. (2022), Reed et al. (2022)), and the field gained attention. First, we will provide an in-depth overview of existing multipurpose models and compare them. In the second part, challenges in the field will also be discussed by reviewing the Pathways proposal (Dean, 2021) and promising work addressing current issues for the progress of multipurpose models.

4.3.1 Prerequisites

At first, we will define the concept of multipurpose models and lay out the necessary prerequisites to make the later described models more accessible.

We will introduce the definition of multipurpose models and further concepts that this book has not covered so far.

4.3.1.1 Multitask Learning

After the extensive overview of multimodal learning in the previous chapter, we now need to introduce multitask learning as another concept to define multipurpose models.

Multitask learning (Caruana (1997), Crawshaw (2020)) is a paradigm in machine learning in which models are trained on multiple tasks simultaneously. Tasks are the specific problems a model is trained to solve, like object recognition, machine translation, or image captioning. Usually, this happens using a single model, which does not leverage helpful knowledge gained from solving other tasks. It is assumed that different tasks include similar patterns that the model can exploit and use to solve other tasks more efficiently. The equivalent in human intelligence is the transfer of knowledge for new tasks since humans do not need to learn each task from scratch but recall previous knowledge that can be reused in the new situation. However, this assumption only sometimes holds since some tasks may require opposing resources, so performance decreases.

Multitask learning thus aims to achieve better generalization by teaching the model how to solve different tasks so that the model learns relationships that can be used further on. For a more in-depth overview of multitask learning, we refer to (Caruana, 1997) and (Crawshaw, 2020).

4.3.1.2 Mixture-of-Experts

Another prerequisite to this chapter is the mixture-of-expert (MoE) (Jacobs et al. (1991), Jordan and Jacobs (1994), Shazeer et al. (2017)) architecture, which is aimed at increasing the overall model size while still keeping inference time reasonably low. In an MoE, not all parts of the net are used but just a subset. The *experts* are best suited to deal with the input allowing the model to be sparse.

MoE is an ensemble of different neural networks inside the layer. MoEs allow for being more computationally efficient while still keeping or even improving performance. The neural networks are not used for every forward pass but only if the data is well suited to be dealt with by a specific expert. Training MoEs usually requires balancing the experts so that routing does not collapse into one or a few experts. An additional gating network decides which of the experts is called. Gating can be implemented so that only K experts are used, which reduces the computational costs for inference by allowing the model to be more sparse.

4.3.1.3 Evolutionary Algorithms

An evolutionary algorithm is used to optimize a problem over a discrete space where derivative-based algorithms cannot be applied to. The algorithm is based on a population (in the domain to be optimized) and a fitness function that can be used to evaluate how close a member of the population is to the optimum. Parts of the population are chosen to create offspring either by mutation or recombination. The resulting population is then evaluated with respect to their fitness function, and only the best-suited individuals are kept. The same procedure is repeated based on the resulting population until a specific criterion is met (e.g., convergence). While evolving the population, it is necessary to balance exploration and exploitation to find the desired outcome. Since EAs are research topics themselves and may vary heavily, we refer to (Bäck and Schwefel, 1993) and, more recently, to (Doerr and Neumann, 2021) for further insights.

4.3.1.4 Multipurpose Models

Now multipurpose models can be defined as multimodal-multitask models. Akin to the underlying assumptions of both learning paradigms, it can also be deduced that multipurpose models mimic human intelligence by marrying the concepts of multiple perceptions and transferring knowledge about different tasks for better generalization.

4.3.2 Overview of Mulitpurpose Models

In this section, we will closely examine existing multipurpose models. The main focus will be on how combining different modalities and tasks is achieved. At the end of this section, all models will be compared to provide a comprehensive overview of promising research directions.

4.3.2.1 MultiModel

The first prominent multipurpose model is the so-called MultiModel (Kaiser et al., 2017). This model, from the pre-transformer era, combines multiple architectural approaches from different fields to tackle both multimodal and multiple tasks. The model consists of four essential modules: The so-called modality nets, the encoder, the I/O Mixer, and the decoder.

Modality nets function as translators between real world data and a suitable representation for the inner modules. They also follow the purpose of back-translating, from the representation to the real world, to create output. For language tasks, the modality net is a tokenizer that outputs the appropriate embeddings, while for vision tasks, convolution operations transform the images into the proper representation. Furthermore, there are also nets for audio and categorical modalities. The modality nets embed the input into a unifying vector space which can be passed to the encoder. To produce the output, the

representations from the decoder are fed into another modality net to produce the output. Language and categories are the only target modalities that have respective modality nets.

The core model consists of the encoder, the I/O mixer, and the decoder. Input is passed from the modality nets to the encoder first. Subsequently, the encoder passes its output further to the I/O mixer and the decoder. The decoder produces the output sequence. However, producing an autoregressive sequence requires knowledge of the previously generated sequence. Thus the output of the decoder is also read by the I/O mixer, which provides the decoder with the necessary information about the previous sequence. The I/O mixer passes its output back to the decoder to provide the necessary information. The decoder and I/O mixer require modality nets to read and write in the target modality. The encoder consists of multiple convolution operations and a mixture-of-expert layer. The I/O mixer and the decoder combine their dual input using cross-attention. A positional encoding conceptually similar to the one in transformers (Vaswani et al., 2017b) is used for the attention mechanism.

MultiModel was trained on eight datasets, from which six were from the language modality and COCO (Lin et al., 2014c) and ImageNet (Russakovsky et al., 2015) from vision. For training, four experts in the MoE layers were used. The combined trained MultiModel on ImageNet and machine translation were below state-of-the-art (SOTA) models. Also, the combined model did not achieve significantly better results than a specialist model, which is the same model but trained solely on one task. However, it was found that the combined model did perform much better on a low-resource task than the respective specialist model.

MultiModel offers a pre-transformer approach to deal with different modalities on multiple tasks; although it is only used to generate text and clasification, the setup allows extending to other modalities easily.

4.3.2.2 Unified Transformer (UniT)

A more recent multipurpose model is UniT (Unified Transformer) (Hu and Singh, 2021b). UniT is built upon the transformer architecture, in which both encoder and decoder are used.

To account for multimodality and multitasking, the basic transformer (Vaswani et al., 2017b) is enhanced. The encoder part of UniT consists of two modality-specific encoders since the initial setup is aimed at the modalities of text and vision. However, more modality-specific encoders may be added. For the case of language, a BERT model (Devlin et al., 2018b) is used, while a detection transformer (DETR) (Carion et al., 2020) encoder is used for vision. DETR uses a particular approach to feed images to the encoder. At first a CNN is used to create a lower dimensional representation of the input image, which is then reorganized as a sequence. This sequence is then fed into the encoder following

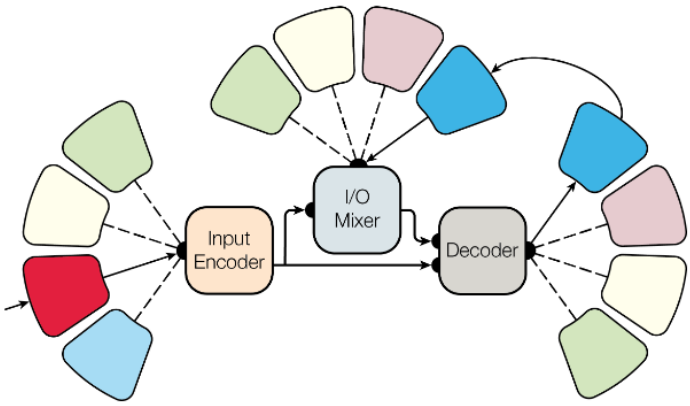


FIGURE 4.15: Architecture of *MultiModel*. The outer boxes without text are the modality nets. From Kaiser et al. (2017).

Vaswani et al. (2017b). The [CLS] token is also used in the BERT encoder, which is also included in the output sequence of the encoder. A task-specific token is additionally added to the input of the encoders. The output of the encoders is then concatenated to form a single sequence. The decoder is fed with this sequence and a task-specific query. Since the decoder architecture sticks to the DETR model, the decoder does not produce a sequence autoregressively. Instead of taking the previously produced sequence (autoregressively) as input, the decoder is fed with the task-specific query vectors instead, thus producing a uniform output. On top of the decoder are task-specific heads needed to transform the decoder output into the desired shape for the specific task.

For training, the object detection task requires bounding box loss from DETR, while the other tasks use cross-entropy loss.

In experiments, UniT was evaluated against a single-task version of itself. The general model outperformed the specialist one on multimodal tasks but was outperformed on unimodal tasks by the specialist UniT. UniT was furthermore also outperformed by SOTA models, although the numbers remained comparable.

Even though UniT does not achieve SOTA or consistently outperforms its specialist version, it is a powerful method to achieve a simple multipurpose model. By using available encoder models, it is easily extendable.

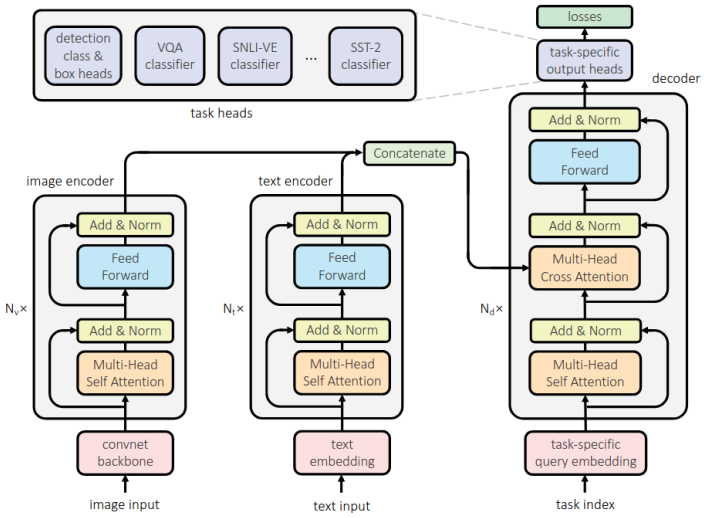


FIGURE 4.16: Modified transformer for UniT. The decoder follows the implementation of DETR (Carion et al., 2020). From Hu and Singh (2021b).

4.3.2.3 OFA - Sequentialization is All You Need

Another multipurpose transformer is OFA (Once For All) (Wang et al., 2022). To utilize the sequence-to-sequence (seq2seq) architecture of the transformer, all input is transformed into a seq2seq problem.

While MultiModel and UniT use specific modules for a modality (modality nets and modality-specific encoders), a different approach is used for OFA. All input is sequentialized and embedded in a shared representation space. Since tokenizing an image using a vocabulary is not feasible, a similar approach to ViT (Dosovitskiy et al., 2020a) is used (where input is flattened to 16 x 16) to obtain a sequence of P representations. These representations are in the same dimension as the token embeddings from text input, which are tokenized using Byte-Pair-Encoding (Sennrich et al., 2016). After feeding the embeddings through the encoder, the decoder produces the output as a sequence again. However, in this case, images are represented as a sequence of tokens, similar to the image-patch vocabulary in DALL-E (Ramesh et al., 2021d). Furthermore, a special sequence for bounding boxes is also used for object detection and recognition. To generate the task-specific solution, it is thus required that another model is used to generate the images based on the tokens and to visualize the bounding boxes based on the obtained coordinates.

Since OFA is an autoregressive model (the probability for the next token is predicted based on the previously produced tokens and the input provided), the objective is based on cross-entropy loss. OFA was trained on different

crossmodal tasks: visual grounding, grounded captioning, image-text matching, image captioning, and visual question answering. Further unimodal tasks for training did include: image infilling, object detection, and text reconstruction as in BART (Lewis et al., 2020).

OFA outperformed SOTA models on cross-modal tasks like image captioning, visual question answering, visual entailment, and visual grounding. On unimodal tasks, OFA performed well, although it did not outperform SOTA models. OFA showed additional transfer capabilities to unseen tasks, which were presented with an additional description to solve the task in a few-shot manner. Although the results were satisfactory, the model was not evaluated against a specialist baseline.

OFA proved to be a powerful model that is capable of using the entire transformer architecture by sequentializing all input and thus producing tokenized output.

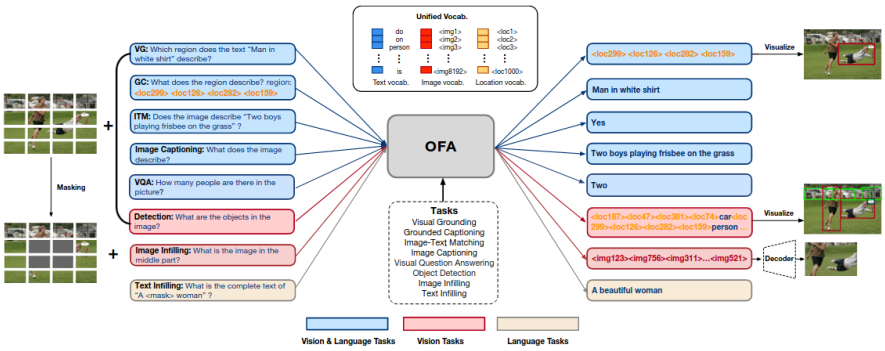


FIGURE 4.17: *OFA*, the different input and output concepts can be seen here. From Wang et al. (2022).

4.3.2.4 Gato - A Generalist Decoder

Another model that utilizes the seq2seq approach in transformers is Gato (Reed et al., 2022). The model can be used as a language model, an agent to play games, and an agent to control robotics.

As in OFA, problems are transformed into a seq2seq problem, on which a transformer (decoder only) is applied. Every input from text, vision, robotics, and games is represented sequentially. Visual input is encoded using a flattened sequence of 16x16 patches fed into a ResNet (He et al., 2015), while text input is tokenized using SentencePiece (Kudo and Richardson, 2018). Furthermore, discrete values like buttons in games and continuous data like movements from robotics are tokenized in a vocabulary too. To represent all modalities sequentially, the different tokens are concatenated. A separator token "|" is

added to distinguish the observations from the following action, so that a sequence looks simplified as the following:

$$\left[\ldots\left[x_{\text{Text}}, x_{\text{Images}}, x_{\text{Discrete and Continuous Values}}, |, y_{\text{Action}}\right]_i, \ldots\right]$$

By using this approach, the transformer can predict the next action autoregressively since it is a sequential problem. In the case of text, the action token is also a text token. Since it is only necessary to predict the action based on the previous values, a mask function is added to the cross-entropy loss function, which masks the previous values so that only the next action is predicted and not the conditions for the action. The masking function is always one for text since every previous text token is necessary for language modeling.

Gato was evaluated on reinforcement-learning-based (RL) tasks against specialist RL agents, where Gato performed worse than the specialist agents. On unseen tasks, Gato required fine-tuning since few-shot learning is not feasible due to the input length restrictions in transformers. However, the results were mixed. Some improvements were possible, and the expert was outperformed, but in other cases, massive fine-tuning efforts only led to small gains. It was found that the generalist agent outperformed the specialist agent (particularly trained for this task) most of the time. Only at the specific Atari Boxing (Bellemare et al., 2013) task, Gato was outperformed by the specialist Gato model. Both performed much lower than another task-specific model used as a baseline. In robotics, Gato showed comparable behavior to the baseline SOTA model. Additionally, Gato also showed capabilities in image captioning and dialogue modeling, although these aspects were not elaborated further.

Like OFA, Gato can sequentialize all input and produce a sequential output that can be back-transformed to solve a task. It was shown that Gato could sometimes transfer knowledge on unseen tasks and outperform the specialist agent most of the time.

4.3.2.5 Comparison

Although many tasks and modalities lead to a curse of dimensionality for comparison, the architectures and the respective modifications of the introduced systems remain simple to compare.

A trend toward seq2seq models can be seen with MultiModel, OFA, and Gato solving tasks in a seq2seq manner. The most prominent similarity is the transformer architecture used entirely (encoder & decoder) in OFA and truncated (decoder only) in Gato. Another significant similarity between both architectures is the use of a particular ordering of input and output. In Gato, the sequence is organized around predicting an action using a special token, while OFA produces a sequence as a solution which can be the bounding box or the sequence of an image to be fed in the generator module. While Gato can solve tasks from robotics and game playing, OFA can also generate images.

However, both architectures require specific modules to decode the tokens into the respective modality.

Gato and OFA both use a shared representation space. Minor details differ, so the image tokenization process is different, and additionally, Gato can encode more modalities than the published version of OFA (although extending OFA is theoretically simple).

MultiModel also show some familiar characteristics. The architecture is from the pre-transformer age but also brings many characteristics of the transformer architecture, like the use of attention, positional encodings, and encoder-decoder. Since the output in the presented version only produced text or classification separately, there is no need for special orderings used in OFA and Gato. The necessity to produce the modality-specific output in modality nets is similar to the generator module in OFA that produces images. However, the tokens are already produced in an intermediate step in OFA, while the modality nets are crucial to producing the final output in MultiModel. UniT follows an entirely different approach that is more pragmatic by leveraging the contextual capabilities of the transformer decoder. M modalities can be encoded as a sequence on which the transformer decoder fuses the modalities and learns the relationships. The use of special tokens for each task and task-specific heads, focus the model on the requested task yet also requires tuning the model specifically.

None of the models besides OFA achieved SOTA results. Compared to specialist models, the general models were comparable in their results (Gato, UniT, MultiModel). MultiModel, OFA, and Gato showed transferability on low-resource or unseen tasks. However, more research in this direction is highly recommended. MultiModel was only compared on a low-resource task against a specialist model, and OFA was not compared to another model for the unseen task. Gato performed better than a specialist model, trained from scratch on most unseen tasks, but failed against the untrained specialist model in Atari Boxing.

Model	Approach	Modalities	Outperformed Specialist Model?	Unseen Tasks?	Number of Parameters	Year
OFA	Seq2Seq	Vision, Text		Yes	33M-930M	2022

Model	Approach	Modalities	Outperformed Specialist Model?	Unseen Tasks?	Number of Parameters	Year
Gato	Seq2Seq	Vision, Text, Robotics, Discrete Entities (e.g., Buttons)	In most cases	Yes	79M-1.18B	2022
UniT	*m* Encoders, task-specific head	Vision, Text	No	No	201M	2021
MultiModel	Different *modality nets* for Seq2Seq	Vision, Text, Audio, Categorical	Comparable	Excelled on low resource task	Unknown	2017

Comparing the models among each other becomes difficult with more modalities and tasks, which is its own curse of dimensionality. For example, Gato also included robotics and RL, which none of the other models included. MultiModel also has a modality net for sound, while UniT and OFA only worked for vision and text. Further research into the comparability of multipurpose models becomes essential.

4.3.3 Pathways and Promising Works

Although models have become more capable of solving complex tasks, significant limitations remain. A persisting issue in current deep learning is the necessity to train from scratch and disregard already obtained knowledge, which is highly ineffective compared to human intelligence. Another issue arises from the evergrowing, dense networks that requires more and more resources.

In this section, we will review the Pathways proposal (Dean, 2021) and promising techniques to address these issues. Overcoming these problems would be especially beneficial for multipurpose models. Reusability of knowledge is crucial for the multitask perspective, and improving the performance of potentially billion-parameter-sized models will also have a significant positive impact.

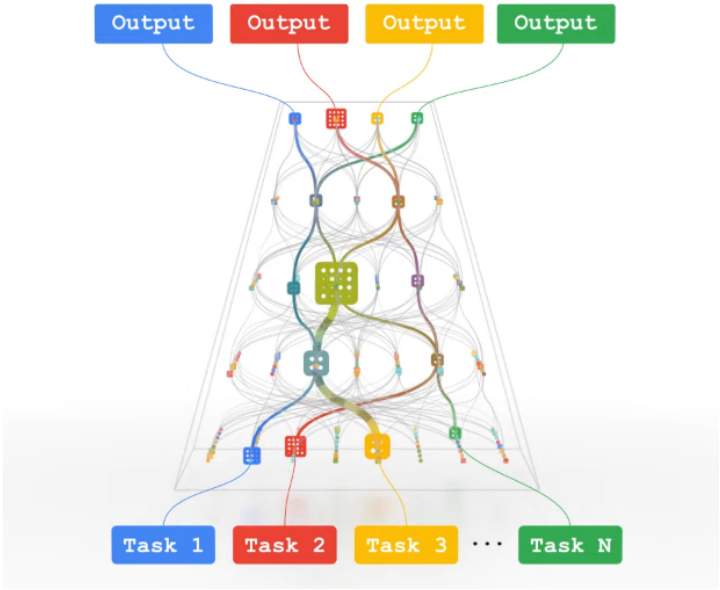


FIGURE 4.18: Concept of Pathways. Different tasks follow different paths to different expert models. From Dean (2021), Screenshot August 31th 2022.

4.3.3.1 Pathways Proposal

Pathways (Dean, 2021) follows a different idea than previously seen methods. The model consists of a large graph through which data can be forward passed. The nodes of the network are neural networks themselves. A pass through this network does not include passing all nodes and thus not all neural networks, but only a few. The pass follows a specific path from one entry to the network's exit. The underlying idea behind this is similar to the mixture-of-expert models described previously. Only the specific networks dedicated to solving a problem are to be activated during inference.

At this point, it is necessary to recall that multitask learning aims to generalize better on new tasks since the knowledge about previously learned tasks can be applied. This idea is the foundation of Pathways too, where specialist networks (nodes) are combined in a larger network. It is assumed that the model's generalization capabilities increase significantly by finding an appropriate path for a task to the appropriate expert nodes. In this setup, the particular task-specific problem-solving capabilities are combined. Furthermore, multimodality is also considered as a potential extension. Adding more modalities might not be a difficult problem considering the architecture of the previously introduced transformer-based models. Overall the approach of a sparse model combining multiple experts offers many opportunities to combine modalities and reuse

task-specific capabilities. The sparsity of the model offers decreased inference time since only few parts of the networks are activated during inference.

Another aspect of the Pathways proposal includes the improvement of current hardware limitations. It is already observable that Moore's Law (*each n years, the compute capacity doubles*) has been slowing down substantially, while deep learning research has grown exponentially in the late 2010s (Dean, 2020). Thus, hardware also needs to be adapted to the growing demand in deep learning. In the context of the pathway proposal, a novel framework for Google data centers has been introduced, aiming to reduce overhead during computation and access specific parts of the model to utilize the technical advantages of sparse networks. As opposed to dense models where a whole model must be accessed, with sparse networks it is not necessary to use the whole network but only chunks of it. So far, two large pre-trained models have been introduced based on the new training framework. One is the Pathways Language Model (PaLM) [Chowdhery2022], which is currently the largest language model using 540 billion parameters, Minerva (Lewkowycz et al., 2022). Minerva is based on PaLM, and Parti (Yu et al., 2022a),

4.3.3.2 PathNet

An earlier approach for a sparse multitask network, which looks deceptively similar, is PathNet (Fernando et al., 2017). PathNet is a training concept that reuses knowledge from a previously learned task without the risk of catastrophic forgetting (knowledge is overwritten), thus using solely the positive aspects of multitask learning. The objective of PathNet consists of a evolutionary algorithm (EA).

Neural networks are often depicted as a graph in which the input is directed to all nodes in hidden layers, and their output is again passed to all nodes in the next hidden layer or an output layer. In the case of PathNet, each node is itself a neural network. The training algorithm finds the best paths for a specific task through the network.

At first random paths through the network are initialized, then the paths are trained for T epochs. After training, the paths are evaluated against each other. The winning path overwrites the losing path. However, to achieve exploration, the overwritten path is mutated by randomly including neighbors of the winning path. Until a specific criterion to stop (e.g., number of epochs) is reached, the current paths are frozen so that no more modifications to the parameters of the networks on this path are possible. All other parameters are newly initialized again. Also, a different, task-specific head is initialized. The same procedure is now done again for the next task. Then, the main difference is that the previously obtained path, including the trained networks, is frozen during training so that the model can transfer knowledge from the previous task to the new task. The model then finds appropriate paths throughout the network until the stopping criterion is met again.

PathNet was evaluated on supervised learning tasks and RL scenarios. Learning from scratch and fine-tuning a PathNet, were chosen as a baseline. For fine-tuning, the first path was chosen as a base model that was fine-tuned on the second task. Overall, PathNet improved training time and prediction quality for the second task compared to standard fine-tuning and learning from scratch. PathNet has shown that different tasks can reuse the knowledge from training on previous tasks without suffering from catastrophic forgetting.

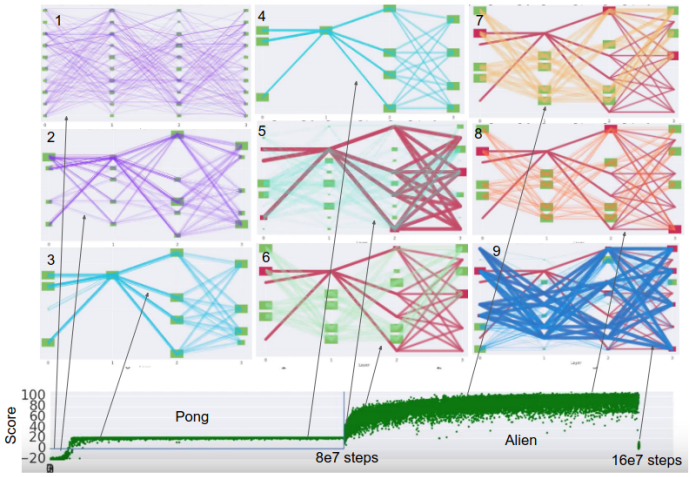


FIGURE 4.19: Training *PathNet* on two tasks. At first random paths are initialized (1), then trained (2-3) and fixed (4). The same procedure is repeated for the next paths using the previously fixed paths and new parameters in all other nodes (5-9). From Fernando et al. (2017).

4.3.3.3 LIMoE

LIMoE (Multimodal Contrastive Learning with LIMoE: the Language-Image Mixture of Experts) (Mustafa et al., 2022) combines text and vision input using a MoE-enhanced transformer encoder.

While previous methods used two models (two-tower) to encode modalities, LIMoE is solely based on one model, where the modalities are processed in a single modified transformer-model (one-tower). The text data is encoded using One-Hot-SentencePiece (Kudo and Richardson, 2018) encoding, while images are tokenized in the same way as in ViT (Dosovitskiy et al., 2020a) (elaborated further in the previous chapter) to provide the input appropriately. The main difference to the standard transformer is an MoE layer where the feed-forward network usually lies. In this layer, E experts are used, which are themselves feed-forward-networks. For each token, K appropriate experts will map the tokens further downstream. The routing is computed by a gating net network, which decides which K experts are called. Another feature here is a

fixed-length buffer for each expert in the MoE layer. This buffer is used to store tokens before an expert network processes them, assuming that the allocations of tokens for each expert are balanced. If it is impossible to buffer tokens for the experts, the tokens will be dropped. To process the more important tokens first, Batch Priority Routing (Riquelme et al., 2021) is used to provide a ranking mechanism. The output of the transformer encoder is then average pooled and subsequently multiplied with a modality-specific weight matrix, which produces the eventual output for the token of both modalities.

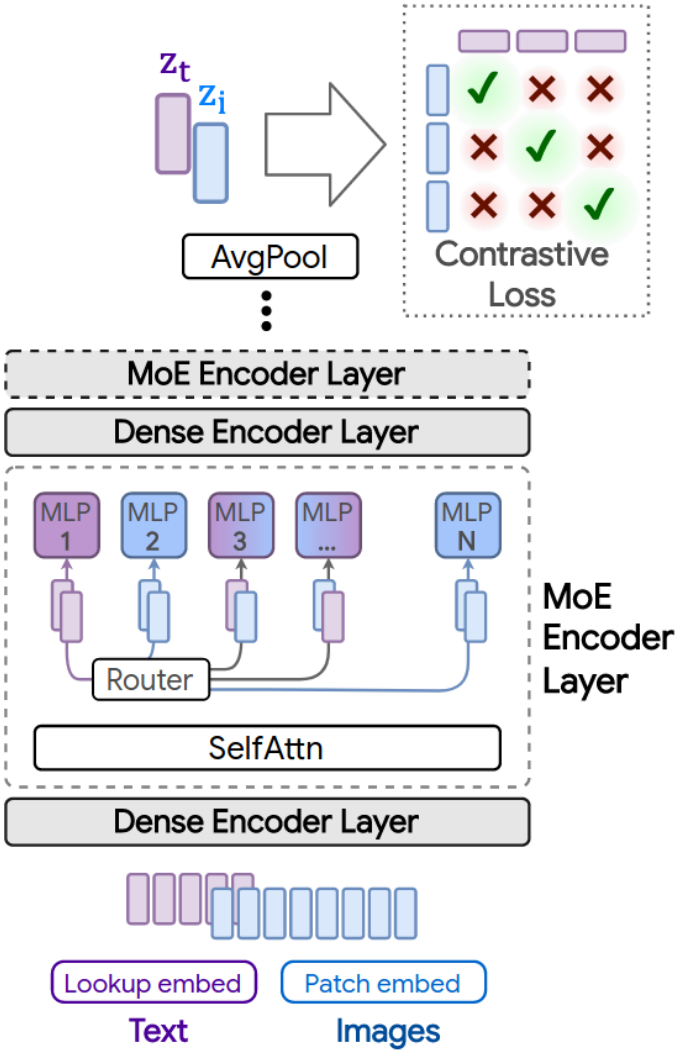


FIGURE 4.20: Architecture of *LIMoE*. From Mustafa et al. (2022).

The model is trained using a contrastive objective. In this case, the contrastive loss aims to maximize the paired visual and textual input while minimizing all combinations of unpaired embeddings. This objective can be achieved by using the dot-product as a similarity measure between the embeddings of both modalities, which provide a differentiable operation through which the overall loss can be minimized.

Additionally, the pitfalls of a multimodal MoE are also considered. One challenge in MoE is the correct balancing of routing to the experts, which is even more challenging when using unbalanced multimodal data. To address this issue, two new losses based on entropy are introduced. Entropy can be used as an appropriate term since it provides a valuable number for the uniformity of the distribution, which is necessary to balance the expert assignments. The losses are aimed at controlling the allocation of experts to tokens, which is also necessary to fulfill the assumptions for the implemented buffer. One loss considers the token level (local) routing distribution, and the other considers the overall expert routing distribution (global). The local loss aims to achieve no uniform behavior in expert allocation such that each token is indeed assigned to specific experts. In contrast, the overall global loss aims to achieve uniformity over all tokens to avoid a collapse in which tokens are solely assigned to a few experts which do not have the capacity to deal with all tokens. These losses are computed for each modality. Furthermore, already available losses for training MoE models were also added to avoid known downsides of MoE models.

LIMoE was compared against similar models like CLIP (Radford et al., 2021a). The test dataset was ImageNet (Deng et al., 2009) and COCO (Lin et al., 2014c). Overall, LIMoE-H/14 (largest model, 12 MoE-layers, 32 experts per layer) achieved strong performance considering that only one model was used for two modalities against specialist models in two-tower setups. It was also possible to outperform CLIP by a significant margin while using minimal additional parameters. Models that achieved similar results to LIMoE used at least twice the number of parameters for a forward pass.

LIMoE provides an example that an MoE-based model achieves impressive results in a multimodal model. Current language and vision encoding techniques are combined and married with the upsides of the MoE-architecture, leading to a single model that can outperform current state-of-the-art models like CLIP.

4.3.3.4 muNet (Multitask Network)

muNet (Gesmundo and Dean, 2022) is an architecture that maximizes the reusability of previously learned knowledge by using an evoluationary algorithm to evolve a new model. The authors address the current practice for fine-tuning, where a pre-trained model is copied and then explicitly trained on a task by overwriting previous knowledge.

An initial model is evolved by using an evoluationary algorithm to fit specific tasks, while keeping the previously learned knowledge. Eventually, a set of models is obtained, which includes new neural networks, based majorly on the parameters of the initial model. The new modules can be seen as paths to task-specific modifications of the initial network.

The EA of muNet starts with an initially proposed model that is mutated further on. All further mutations are stored so that after a set of candidates is available, the set can be split into models trained for this task (active population) and models for other tasks (inactive population). These two sets become the sets of candidates for the following task-specific iterations. Training a specific task follows three steps: Sampling candidate models, mutating, training, and evaluation. The best scoring model is added to the active population for further mutation. A sampling algorithm accounts for exploration and exploitation to get a candidate model for subsequent mutation. The active population is ordered in a descending list based on the model's score. Each list entry is then revisited, starting from the highest scoring model onward, so that the better performing models are considered first (exploitation). The draw probability is computed as:

$$\mathbb{P}(m|t) = 0.5^{\#timesSelected(m,t)}$$

Where $\#timesSelected(m,t)$ is the amount of previous mutations based on model m for task t). The more unsuccessful mutations the model has had before, the smaller the draw probability becomes. Thus, exploration is emphasized by considering previous attempts and allowing other models to be preferred as well. However, if this method does not yield a candidate, a model is drawn from the union of the inactive and active population. Applying mutations is the next step in the algorithm. A random number of mutations are drawn from the set of possible mutations, which include:

- **Layer Cloning**: A layer is cloned for training. The layer's parameters are copied from the parent model so that training can continue using the same knowledge. The other layers are still used but are not updated. Additionally, the task-specific head layer is cloned to account for the underlying changes. In case of training on a new task, the head is also newly initialized.
- **Layer Insertion**: Two layers are added to the model as residual adapters ((Rebuffi et al., 2017), [Houlsby2019@]). The second layer is zero-initialized to keep an identity function so that training can continue from the state before mutation.
- **Layer Removal** is used to skip layers while still using all other layers of the parent model in a frozen state.
- **Hyperparameter Change**: samples hyperparameters close to the ones of the parent model. A list of neighboring values is constructed from which a parameter is drawn.

Subsequently, the models are trained on the task and scored. If the mutated

model is better than the parent model, it is also added to the task's set of active models. This routine is done for all tasks iteratively and can be repeated several times. Ultimately, only the best scoring models are kept for each task, yielding a list of models each fit to a particular task. muNet was evaluated for fine-tuning against a ViT instance, which was aimed at being the most generalizable one (Steiner et al., 2021). The evaluation benchmarks consisted of multiple classification problems (to simulate multitasking). ViT was fine-tuned on all of these tasks as a baseline. In contrast, another ViT was evolved using muNet, on which the baseline model was evaluated again. The approach using muNet outperformed the fine-tuned ViT while using significantly fewer parameters.

muNet offers a simple, evolutionary-based approach for fine-tuning and keeping all previously acquired knowledge safe, thus maximizing reusability.

4.3.4 Conclusion Pathways

The introduced models show promising novel features that might improve multipurpose models. However, these models can only be improved if research is done to combine the distinct concepts. PathNet and muNet offer novel approaches to leverage already acquired knowledge, while LIMoE improves handling different modalities in a single, sparse model. Furthermore, it also becomes necessary to conduct research into scaling these concepts up. Since the multitask-related models (PathNet and muNet) only included a few tasks, introducing more tasks for training and testing might offer insights into how transfer between tasks succeeds and fails.

LIMoE offers a promising architecture with respect to performance. Due to the sparsity of the MoE-layer, LIMoE is faster, while it also outperforms previous dense models. Using MoE-layers in transformers might also be a viable path for models like OFA and Gato. Combining the flexible encoding techniques of these models with the relative sparsity of LIMoE might result in even more capable and efficient models. We, therefore, recommend further research in this direction.

Another potential path for future research is intelligent routing for evolving methods like muNet and PathNet. Evolutionary models offer a promising approach to leveraging previous knowledge. However, the resulting models are tailored to a particular task. Novel routing techniques to send data to dedicated expert nodes in a complex network of models might help models generalize, as was outlined in the Pathways proposal.

4.3.5 Discussion

We reviewed multipurpose models that have become capable of solving multiple tasks from different modalities. The transformer architecture also boosted

the development in this field, in which three of the four presented models were transformer-based and from recent years. Multipurpose models offers an opportunity to use one model instead of many different expert-models. Furthermore, some multipurpose models (Gato, OFA) also outperformed expert-models. However, Gato also showed inferior performance on ATARI Boxing compared to competing models, indicating that research is still required to explore the relationship between tasks. We also presented promising novel architectures that alleviate or may solve problems in current multipurpose models. However, further issues remain that have not been solved by research to this day:

- A pitfall of models of these sizes is the low accessibility. Researchers need to access the model through an API since running these models on a few GPUs will likely be infeasible. It might be unlikely to see a BERT-like engagement with the community of researchers if the access to models remains limited. On the contrary, more open-source collaborations, as seen with EleutherAI or Huggingface, might evolve as well as a countermovement and techniques like distillation (Hinton et al., 2015a) might become more critical.
- Another issue with multipurpose models is the lack of metrics. Current metrics are not suited for multitask and multimodal models. Evaluation might also become harder since many different modalities can be used, as seen here with the robotics property of Gato, which was not used in any of the other reviewed models.
- Eventually, it is also necessary to consider the societal impact. The bias problem will also become an issue in multipurpose models, especially since multiple datasets must be considered.
- Also, the environmental impact of training large models needs to be considered since it is likely that larger models will yield better performance according to scaling laws (Reed et al., 2022) but will also have a larger carbon footprint.

4.4 Generative Art

Author: Nadja Sauter

Supervisor: Jann Goschenhofer

As we have seen in subsection 3.2, computers can create images only based on text prompts via multimodal deep learning. This capability is also used in digital arts in the field of 'generative art' or also known as 'computer art'. The new movement comprises all artwork where the human artist cedes control to an autonomous system (Galanter, 2016). In this way everyone, even artistically

FIGURE 4.21: LMU logo in style of Van Gogh's Sunflower painting

untrained people, can easily create pictures as the computer takes over the image generation. In some way, the computer becomes the artist with some sort of creativity, a distinct human ability. In this chapter, we want to give an overview about how computers improved over time in generating images and how this is used in the contemporary arts scene. For instance in Figure 4.21 we used the seal of the Ludwig Maximilians University and changed the style to Van Gogh's Sunflower painting by the Neural Stlye Transfer Algorithm and the method CLIP + VQGAN which fuses the logo with sunflowers in a Van-Gogh-style way.

4.4.1 Historical Overview

The first attempt to use AI to generate pictures was made by the engineer Alexander Mordvintsev (2015) and his "DeepDream" Software. He used Convolution Neural Networks to generate very interesting and abstract images based on the activation of a layer, visualizing the patterns learned by a neural network. Below you can see a picture of a Labrador after it was processed by the DeepDream algorithm.

In the following year, Gatys et al. (2016) investigated methods to transfer the style of pictures. This method was used to transfer the style of Van Gogh's Sunflower painting to the LMU seal at the beginning of this chapter (see Figure 4.21). Besides, below in Figure 4.23 you can see the same Labrador picture from Figure 4.22 in Kandinsky style.

Furthermore, the architecture of Generative Adversarial Networks (GANs), which was first introduced by Goodfellow et al. (2014a), was used by another research group Karras et al. (2019) to create very realistic fake images with their architecture StyleGAN. For instance, one can create pictures of people who do not exist, but look totally realistic (see Figure 4.24).

Nevertheless, it was almost impossible to control the exact output of these early forms of AI art. There was no option to make specifications of how the

FIGURE 4.22: Picture of a Labrador processed by DeepDream (Google Colab)

FIGURE 4.23: Picture of a Labrador with Kandinsky style (Google Colab)

FIGURE 4.24: Fake face generated by StyleGAN

result should look like in detail. For instance, you always get a human face with the earlier mentioned StyleGAN application, but you cannot specify to generate a blond girl with green eyes. This can be achieved by applying the artist-critic paradigm (Soderlund and Blair, 2018): Thereby, the computer as an artist generates a picture based on what the Neural Network learned in the training phase (e.g. StyleGAN learns to generate pictures of human faces). Additionally, a critic is used to tell the computer if the output satisfies the concrete idea of the human artist. For this reason multimodal deep learning models emerged in the field of generative art. Here, one can control the output with the help of text prompting. In this way one can check if the generated picture matches the initial text description. Looking at the previous StyleGAN example, the multimodal architecture supervises whether the output picture is indeed a blond girl with green eyes or not. A new class of models for generating pictures evolved.

This idea was used by OpenAI for their models DALL-E (Ramesh et al., 2021a) and CLIP (Radford et al., 2021b) which were released in January 2021. Both architectures are critics for multimodal models. Only a few days after the release, Ryan Murdock combined CLIP (critic) with the already existing Neural Net "BigGAN" (artist) in his "The Big Sleep" software. Furthermore, Patashnik et al. (2021) developed StyleCLIP, a combination of StyleGAN (artist) and CLIP (critic) to edit parts of images via text instructions. In the following months, Katherine Crowson combined CLIP as critic with the existing VQGAN algorithm as an artist. She also hooked up CLIP with guided diffusion models as artists to yield more fine-grained results. This approach was further investigated by OpenAI that published a paper (Dhariwal and Nichol, 2021) in May 2021 about guided diffusion models. Moreover, in December 2021 they introduced GLIDE (Nichol et al., 2021a), a model with CLIP or classifier-free guidance as critics and diffusion models as artists. For more technical details about text2img methods like DALL-E and GLIDE refer to subsection 3.2 or for text supporting CV models like CLIP at subsection 3.4.

4.4.2 How to use these models?

A lot of different notebooks are publicly available to apply the different pre-trained models. In general, all notebooks work pretty similar: one only needs to enter a text prompt in the code and after running the notebook the computer generates a picture based on these instructions. It is relatively easy and no prior coding knowledge is required. Moreover, there are also some API and GUI applications (e.g. MindsEye beta) where no programming knowledge is needed at all. Using these models, it is important to think about how exactly once enters the respective text prompt. One can influence the output in a desired way with little changes in the short text instruction. This is also known as "prompt engineering". For instance, in the beginning of this chapter, we entered the prompt "in the style of Van Gogh" to change the style of the LMU

seal. In this context, a special trick is to append "unreal engine" (Aran, 2021) which makes the resulting pictures more realistic with higher quality. This seems surprising at first, but the models were trained on data from the internet including pictures of the software company Epic Games that has a popular 3D video game engine called "Unreal Engine". This is one of the most popular prompting tricks.

Unfortunately, OpenAI has never released DALL-E. There is only an open-source version called ruDALL-E (Shonenkov, 2021) that was trained on Russian language data. Besides, hugging face hosts DALL-E mini (Boris, 2022) where one can generate pictures, but does not have access to the model itself. PyTorch offers a replication of the DALL-E code (OpenAI, 2021) but no trained model. Furthermore, CLIP was released without publishing the used training data. However, there exists an open source data set with CLIP embeddings called LAION-400m (Schuhmann et al., 2021b). In the following, we used different publicly available notebooks to try out the different models CLIP + BigGAN, CLIP + VQGAN, CLIP + Guided Diffusion, GLIDE with the text prompt *"a fall landscape with a small cottage next to a lake"* (see Figure 4.25) and *"panda mad scientist mixing sparkling chemicals, artstation"* (see Figure 4.26). The first prompt shows pretty realistic results, whereas the second prompt results in more different and "crazy" outputs. That is because the panda-prompt is more abstract than the first one and hence more difficult to illustrate. In addition, some of the notebooks run on lower resolution due to computational limitations. Besides, GLIDE is also downsized by the publisher: The released smaller model consists of 300 million parameters, whereas the unreleased model has about 3.5 billion parameters (Nichol et al., 2021a). So better results are possible with higher computational power and other implementations of the models.

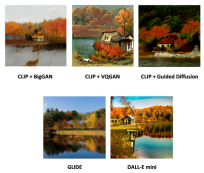


FIGURE 4.25: Comparison of different models with prompt "fall landscape with a small cottage next to a lake"

4.4.3 Different tasks and modalities

So far, we concentrated on the two modalities text and image. Combining both of them, one can tackle different tasks with the models mentioned above. The main usage is to generate images based on a text prompt. Therefore, one can start from noise or but is also possible to chose a real image as starting point (Qiao et al., 2022). This was done in the beginning with the LMU seal by CLIP

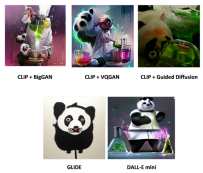


FIGURE 4.26: Comparison of different models with prompt "panda mad scientist mixing sparkling chemicals, artstation"

+ VQGAN (see Figure 4.21): instead of starting from noise, the model started from the LMU seal as initialization and then used the prompt "in style of Van Gogh". The video captures how the model develops during fitting. In the end, the typical Van Gogh sunflowers emerge as well as what could be a part of Van Gogh's face.

Furthermore, one can edit, extend, crop and search images with models like GLIDE (Nichol et al., 2021a). For instance, Nichol et al. (2021a) fine-tuend the model for text-conditional image inpainting (see figure 4.27). By marking some area in the pictures, here in green, and adding a text prompt, one can edit pictures very easily and precisely. This is quite impressive as the model needs to understand from the text prompt which object should be filled in and then do this in the correct style of the surrounding to produce a realistic outcome. Another idea is to use a sketch of a drawing and let the model fill in the details based on a text caption (see figure 4.28 below). This allows controlled changes of parts of pictures with relatively little effort. In this way, GLIDE can be used to generate pictures out of random noise, but also to edit pictures in a specific way. Furthermore, it is also possible to combine other modalities as well (see more details in subsection 4.1). For instance, WZRD (2020) accompanies custom videos with suitable audio. It is even imaginable to create sculptures with 3D-printers (Mccormack and Gambardella, 2022).

FIGURE 4.27: Text-conditional image inpainting examples with GLIDE (Nichol et al., 2021a)

"a corgi wearing a bow tie and a birthday hat"

FIGURE 4.28: Text-conditional edit from user scratch with GLIDE (Nichol et al., 2021a)

4.4.4 Discussion and prospects

In the last years, methods to generate images via text prompting improved tremendously and a new field of art arised. It is surprising how these models are able to create images only based on a short text instruction. This is quite impressive as AI achieved some level of creativity. It is up for discussion to which extent the computer is becoming the artist in generative arts and in this way replacing the human artist. However, there is still no direct loss function that can calculate how aesthetically pleasing a picture is (Esser et al., 2020). This is probably also quite subjective and cannot be answered for everyone in the same way. Most of the time the computer works as aid for the creative process by generating multiple images. Then, the human artist can pick the best outcome or vary the text prompt to improve the output in a desired way. However, the better the AI becomes, the less the human artist needs to intervene in this process.

Furthermore, as the output becomes more and more realistic, there is the risk that these methods are abused to facilitate plagiarism or create fake content and spread misleading information (Dehouche, 2021). After all, the outputs look totally realistic, but are completely made-up and generated by the computer. For this reason, some organisations like Open-AI do not release all their models (e.g. DALL-E) or downstream models (e.g. CLIP). On the other hand, from a scientific point of view, it is important to get access to such models to continue research.

Moreover, similarly to most Deep Learning algorithms, these models are affected by biases in the input data (Srinivasan and Uchino, 2021). For instance, Esser et al. (2020) points out that CLIP text embeddings associate a human being more with a man than with a woman. In this way it might be more likely that our models generate a man with the text prompt "human being" than a woman. This effect needs to be further investigated and should be removed.

After all, generative arts can be used to create Non Fungible Tokens (NFT) relatively easily. NFTs are digital artworks where a special digital signature is added making them unique and in this way non-fungible (Wang et al., 2021). The digital artwork is bought and sold online, often by means of cryptocurrency. That is why this field is also called Cryptoart. This provides the perfect platform to sell generative arts. However, this trading market is quite new and controversial, similar to crypotcurrency trading in general.

In conclusion, generative arts is a new and impressive field. It combines technology with arts, two rather opposite fields. The methods are already really impressive and are still getting better and better. For instance, this year Open AI already published DALLE-2 (Ramesh et al., 2022a) that outperforms DALLE-1. It remains highly interesting to follow up with the developments in this field.

5

Conclusion

Author: Nadja Sauter

Supervisor: Matthias Aßenmacher

It is very impressive how multimodal architectures have developed, especially over the course of the last two years. Particularly, methods to generate pictures based on text prompts, like DALL-E, became incredibly good at their "job". A lot of people are fascinated by the stunning results and a huge hype about these AI generated images evolved in the internet, especially on twitter. In this way, the models were not only investigated by researchers but also by the online community (e.g. Katherine Crowson alias Rivers Have Wings). Even in the art scene these methods attracted a lot of attention as shown in our use case "enerative Arts" (subsection 4.4). Apart from that, it is possible to deploy these methods commercially, for instance in the film production or gaming industry (e.g. creating characters for games). However, this might also result in problems of copyright, an issue which has not yet been dealt with until now.

It is also impressive how realistic and precise outputs are achieved by such architectures. On the other hand, these methods can also be abused to spread misleading information as it is often very difficult to distinguish between a fake or a real picture by only looking at it. This can be systematically used to manipulate the public opinion by spreading AI manipulated media, also called deep fakes. That's why researchers like Joshi et al. (2021) demand automated tools which are capable of detecting these fabrications. Apart from that, like most deep learning models, multimodal architectures are not free from bias which also needs to be investigated further (Esser et al., 2020). Besides, the algorithms are very complex which is why they are often called "black-box" models, meaning that one cannot directly retrace how the model came to a certain solution or decision. This may limit their social acceptance and usability as the underlying process is not credible and transparent enough (Joshi et al., 2021). For instance, in medical applications like e.g. predicting the presence or absence of cancer, apart from the decision of the AI the reasoning and the certainty are highly relevant for doctors and patients.

Furthermore, there is a clear trend in recent years to build more and more complex architectures in order to achieve higher performance. For instance OpenAI's language model GPT-2 has had about 1.5 billion parameters (Radford et al., 2019a), whereas its successor GPT-3 had about 175 billion parameters

(Brown et al., 2020). Increasing the number of parameters often helps improving model performance, but all of these parameters need to be trained and stored which takes a lot of time, enormous computational power and storage. For example, training GPT-2 took about one week (168 hours) of training time on 32 TPUv3 chips (Strubell et al., 2019c). The researchers (Strubell et al. (2019c)) estimated that the cloud compute costs for training GPT-2 added up to about \$12,902–\$43,008. Apart from the enormous expenses, this also contributes to our environmental burden as this process is really energy intensive. Due to missing power draw data on GPT-2's training hardware, the researchers weren't able to calculate the CO_2 emission. However, for the popular BERT architecture with 110M parameters they calculated cloud compute costs of \$3,751-\$12,571, energy consumption of 1,507 kWh and a Carbon footprint of 1,438 lbs of CO_2. In comparison, the footprint of flying from New York to San Francisco by plane for one passenger is about 1,984 lbs of CO_2. In conclusion training BERT once results in almost the same footprint as this long-haul flight. On top of this, these numbers are only for one training run. Developing a new model or adapting it often takes several fitting and tuning phases.

Moreover, the computational power as well as the necessary hardware, technology and financial means to run these models can oftentimes only be provided by big technology companies like e.g. Google, Facebook or OpenAI. This results in a disparate access between researchers in academia versus industry. Furthermore, the companies sometimes tend do not publishing their (best) models as they are their "product" and contribute to the company's intellectual property. In this way it is not possible to reproduce their work and findings independently. Besides, from an economic point of view, this may be the foundation of a monopoly which might be dangerous for economic competition and holds the possibility of abuse.

6

Epilogue

Author: Matthias Aßenmacher

Since this project was realized in a limited time frame and accounted for about one third of the ECTS points which should be achieved during one semester, it is obvious that this booklet cannot provide exhaustive coverage of the vast research field of *Multimodal Deep Learning*.

Furthermore this area of research is moving very rapidly at the moment, which means that certain architectures, improvements or ideas had net yet even been published when we sat down and came up with the chapter topics in February 2022. Yet, as you might have seen, in some cases the students were even able to incorporate ongoing research published over the course of the seminar. Thus, this epilogue tries to put the content of this booklet into context and relate it to what is currently happening. Thereby we will focus on two aspects:

- New influential (or even state-of-the-art) architectures
- Extending existing architectures to videos (instead of "only" images)

6.1 New influential architectures

In **Chapter 3.2: "Text2Image"** and **Chapter 4.4: "Generative Art** some important models for generating images/art from free-text prompts have been presented. However, one example of an even better (at least perceived this way by many people) generative model was just published by researchers from Björn Ommer's group at LMU: *"High-Resolution Image Synthesis with Latent Diffusion Models"*
They introduced a model called *Stable Diffusion* which allows users to generate photorealisitic images. Further (as opposed to numerous other architectures, it is available open-source and can even be tried out via huggingface.

6.2 Creating videos

Also more recently, research has focussed on not only creating images from natural language input but also videos. The *Imagen* architecture, which was developed by researchers at Google Research (Brain Team), was extended with respect to also creating videos (see their project homepage). Yet, this is only on of many possible examples of research being conducted in this direction. The interested reader is refered to the paper accompanying their project.

We hope that this little outlook can adequately round off this nice piece of academic work created by extremely motivated students and we hope that you enjoyed reading.

7

Acknowledgements

The most important contributions are from the students themselves. The success of such projects highly depends on the students. And this book is a success, so thanks a lot to all the authors! The other important role is the supervisor. Thanks to all the supervisors who participated! Special thanks to Christian Heumann and Bernd Bischl who enabled us to conduct the seminar in such an experimental way, supported us and gave valuable feedback for the seminar structure. Thanks a lot as well to the entire Department of Statistics and the LMU Munich for the infrastructure.

The authors of this work take full responsibilities for its content.

Bibliography

(2022). Neural Networks - History. [Online; accessed 2022-06-29].

Agirre, E., Alfonseca, E., Hall, K., Kravalova, J., Pasca, M., and Soroa, A. (2009). A study on similarity and relatedness using distributional and wordnet-based approaches.

Ailem, M., Zhang, B., Bellet, A., Denis, P., and Sha, F. (2018). A probabilistic model for joint learning of word embeddings from texts and images.

Akbari, H., Yuan, L., Qian, R., Chuang, W., Chang, S., Cui, Y., and Gong, B. (2021). VATT: transformers for multimodal self-supervised learning from raw video, audio and text. In Ranzato, M., Beygelzimer, A., Dauphin, Y. N., Liang, P., and Vaughan, J. W., editors, *Advances in Neural Information Processing Systems 34: Annual Conference on Neural Information Processing Systems 2021, NeurIPS 2021, December 6-14, 2021, virtual*, pages 24206–24221.

Alayrac, J.-B., Donahue, J., Luc, P., Miech, A., Barr, I., Hasson, Y., Lenc, K., Mensch, A., Millican, K., Reynolds, M., et al. (2022). Flamingo: a visual language model for few-shot learning.

Alford, A. (2021). Google announces 800m parameter vision-language ai model align.

Anderson, P., Fernando, B., Johnson, M., and Gould, S. (2016). Spice: Semantic propositional image caption evaluation. In Leibe, B., Matas, J., Sebe, N., and Welling, M., editors, *Computer Vision – ECCV 2016*, pages 382–398. Springer International Publishing.

Anderson, P., He, X., Buehler, C., Teney, D., Johnson, M., Gould, S., and Zhang, L. (2018). Bottom-up and top-down attention for image captioning and visual question answering. In *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 6077–6086.

Antol, S., Agrawal, A., Lu, J., Mitchell, M., Batra, D., Zitnick, C. L., and Parikh, D. (2015). Vqa: Visual question answering. In *Proceedings of the IEEE international conference on computer vision*, pages 2425–2433.

Aran, K. (2021). When you generate images with vqgan clip, the image quality dramatically improves if you add "unreal engine" to your prompt. people are now calling this "unreal engine trick".

Bachmann, R., Mizrahi, D., Atanov, A., and Zamir, A. (2022). Multimae: Multi-modal multi-task masked autoencoders.

Baevski, A., Hsu, W.-N., Xu, Q., Babu, A., Gu, J., and Auli, M. (2022). Data2vec: A general framework for self-supervised learning in speech, vision and language.

Baevski, A., Zhou, Y., Mohamed, A., and Auli, M. (2020). wav2vec 2.0: A framework for self-supervised learning of speech representations. 33:12449–12460.

Bahdanau, D., Cho, K., and Bengio, Y. (2014). Neural machine translation by jointly learning to align and translate.

Bandy, J. and Vincent, N. (2021). Addressing" documentation debt" in machine learning research: A retrospective datasheet for bookcorpus.

Banerjee, S. and Lavie, A. (2005). METEOR: An automatic metric for MT evaluation with improved correlation with human judgments. In *Proceedings of the ACL Workshop on Intrinsic and Extrinsic Evaluation Measures for Machine Translation and/or Summarization*, pages 65–72. Association for Computational Linguistics.

Bao, H., Dong, L., and Wei, F. (2021). Beit: Bert pre-training of image transformers.

Barham, P., Chowdhery, A., Dean, J., Ghemawat, S., Hand, S., Hurt, D., Isard, M., Lim, H., Pang, R., Roy, S., Saeta, B., Schuh, P., Sepassi, R., Shafey, L. E., Thekkath, C. A., and Wu, Y. (2022). Pathways: Asynchronous distributed dataflow for ml.

Bellemare, M. G., Naddaf, Y., Veness, J., and Bowling, M. (2013). The arcade learning environment: An evaluation platform for general agents. 47(1):253–279.

Bender, E. M., Gebru, T., McMillan-Major, A., and Shmitchell, S. (2021). On the dangers of stochastic parrots: Can language models be too big? In *Proceedings of the 2021 ACM Conference on Fairness, Accountability, and Transparency*, FAccT '21, pages 610–623. Association for Computing Machinery.

Bengio, Y., Courville, A. C., and Vincent, P. (2013). Representation learning: A review and new perspectives. 35(8):1798–1828.

Beyer, L., Hénaff, O. J., Kolesnikov, A., Zhai, X., and Oord, A. v. d. (2020). Are we done with imagenet?

Birhane, A., Prabhu, V. U., and Kahembwe, E. (2021). Multimodal datasets: misogyny, pornography, and malignant stereotypes.

Bojanowski, P., Grave, E., Joulin, A., and Mikolov, T. (2016). Enriching word vectors with subword information.

Bommasani, R., Hudson, D. A., Adeli, E., Altman, R., Arora, S., von Arx, S., Bernstein, M. S., Bohg, J., Bosselut, A., Brunskill, E., et al. (2021). On the opportunities and risks of foundation models.

Bordes, P., Zablocki, E., Soulier, L., Piwowarski, B., and Gallinari, P. (2020). Incorporating visual semantics into sentence representations within a grounded space.

Boris, D. (2022). Dall · e mini.

Borji, A. (2018). Pros and cons of GAN evaluation measures.

Bosch, A., Zisserman, A., and Munoz, X. (2007). Image classification using random forests and ferns. In *2007 IEEE 11th international conference on computer vision*, pages 1–8. Ieee.

Bowman, S. R. and Dahl, G. E. (2021). What will it take to fix benchmarking in natural language understanding?

Bromley, J., Guyon, I., LeCun, Y., Säckinger, E., and Shah, R. (1993). Signature verification using a" siamese" time delay neural network. 6.

Brown, T., Mann, B., Ryder, N., Subbiah, M., Kaplan, J. D., Dhariwal, P., Neelakantan, A., Shyam, P., Sastry, G., Askell, A., et al. (2020). Language models are few-shot learners. 33:1877–1901.

Bruni, E., Tran, N.-K., and Baroni, M. (2014). Multimodal distributional semantics. 49:1–47.

Brysbaert, M., Warriner, A. B., and Kuperman, V. (2014). Concreteness ratings for 40 thousand generally known english word lemmas. 46(3):904–911.

Bäck, T. and Schwefel, H.-P. (1993). An overview of evolutionary algorithms for parameter optimization. 1(1):1–23.

Carion, N., Massa, F., Synnaeve, G., Usunier, N., Kirillov, A., and Zagoruyko, S. (2020). End-to-end object detection with transformers.

Caron, M., Misra, I., Mairal, J., Goyal, P., Bojanowski, P., and Joulin, A. (2020). Unsupervised learning of visual features by contrasting cluster assignments.

Caron, M., Touvron, H., Misra, I., Jégou, H., Mairal, J., Bojanowski, P., and Joulin, A. (2021). Emerging properties in self-supervised vision transformers. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 9650–9660.

Carreira, J., Koppula, S., Zoran, D., Recasens, A., Ionescu, C., Henaff, O., Shelhamer, E., Arandjelovic, R., Botvinick, M., Vinyals, O., Simonyan, K., Zisserman, A., and Jaegle, A. (2022). Hierarchical perceiver.

Caruana, R. (1997). Multitask learning. *Machine learning*, 28(1):41–75.

Cheerla, A. and Gevaert, O. (2019). Deep learning with multimodal representation for pancancer prognosis prediction. 35(14):i446–i454.

Chen, T., Kornblith, S., Norouzi, M., and Hinton, G. (2020a). A simple framework for contrastive learning of visual representations.

Chen, T., Kornblith, S., Norouzi, M., and Hinton, G. E. (2020b). A simple framework for contrastive learning of visual representations. In *Proceedings of the 37th International Conference on Machine Learning, ICML 2020, 13-18 July 2020, Virtual Event*, volume 119 of *Proceedings of Machine Learning Research*, pages 1597–1607. PMLR.

Chen, X., Xie, S., and He, K. (2021). An empirical study of training self-supervised vision transformers. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 9640–9649.

Cheng, H.-T., Koc, L., Harmsen, J., Shaked, T., Chandra, T., Aradhye, H., Anderson, G., Corrado, G., Chai, W., Ispir, M., et al. (2016). Wide & deep learning for recommender systems. In *Proceedings of the 1st workshop on deep learning for recommender systems*, pages 7–10.

Cho, K., van Merrienboer, B., Gulcehre, C., Bahdanau, D., Bougares, F., Schwenk, H., and Bengio, Y. (2014). Learning phrase representations using rnn encoder-decoder for statistical machine translation.

Chowdhery, A., Narang, S., Devlin, J., Bosma, M., Mishra, G., Roberts, A., Barham, P., Chung, H. W., Sutton, C., Gehrmann, S., et al. (2022). Palm: Scaling language modeling with pathways.

Clark, K., Khandelwal, U., Levy, O., and Manning, C. D. (2019). What does bert look at? an analysis of bert's attention.

Collell, G., Zhang, T., and Moens, M.-F. (2017). Imagined visual representations as multimodal embeddings. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 31.

Cornia, M., Stefanini, M., Baraldi, L., and Cucchiara, R. (2019). Meshed-memory transformer for image captioning.

Cornia, M., Stefanini, M., Baraldi, L., and Cucchiara, R. (2020). Meshed-Memory Transformer for Image Captioning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*.

Crawshaw, M. (2020). Multi-task learning with deep neural networks: A survey.

Crowson, K., Biderman, S., Kornis, D., Stander, D., Hallahan, E., Castricato, L., and Raff, E. (2022). Vqgan-clip: Open domain image generation and editing with natural language guidance.

Da, J. and Kasai, J. (2019). Cracking the contextual commonsense code:

Understanding commonsense reasoning aptitude of deep contextual representations.

Das, A., Agrawal, H., Zitnick, L., Parikh, D., and Batra, D. (2017). Human attention in visual question answering: Do humans and deep networks look at the same regions? 163:90–100.

Dean, J. (2020). 1.1 the deep learning revolution and its implications for computer architecture and chip design. In *2020 IEEE International Solid-State Circuits Conference - (ISSCC)*, pages 8–14.

Dean, J. (2021). Introducing pathways: A next-generation ai architecture.

Dehouche, N. (2021). Plagiarism in the age of massive generative pre-trained transformers (gpt-3). 21:17–23.

Deng, J., Dong, W., Socher, R., Li, L.-J., Li, K., and Fei-Fei, L. (2009). Imagenet: A large-scale hierarchical image database. In *2009 IEEE conference on computer vision and pattern recognition*, pages 248–255. Ieee.

Devereux, B. J., Tyler, L. K., Geertzen, J., and Randall, B. (2014). The centre for speech, language and the brain (cslb) concept property norms. 46(4):1119–1127.

Devlin, J., Chang, M.-W., Lee, K., and Toutanova, K. (2018a). Bert: Pre-training of deep bidirectional transformers for language understanding.

Devlin, J., Chang, M.-W., Lee, K., and Toutanova, K. (2018b). Bert: Pre-training of deep bidirectional transformers for language understanding.

Devlin, J., Chang, M.-W., Lee, K., and Toutanova, K. (2018c). Bert: Pre-training of deep bidirectional transformers for language understanding.

Devlin, J., Chang, M.-W., Lee, K., and Toutanova, K. (2019). BERT: Pre-training of deep bidirectional transformers for language understanding. pages 4171–4186. Association for Computational Linguistics.

Dhamala, J., Sun, T., Kumar, V., Krishna, S., Pruksachatkun, Y., Chang, K.-W., and Gupta, R. (2021). Bold: Dataset and metrics for measuring biases in open-ended language generation. In *Proceedings of the 2021 ACM Conference on Fairness, Accountability, and Transparency*, pages 862–872.

Dhariwal, P. and Nichol, A. (2021). Diffusion models beat gans on image synthesis.

Ding, M., Yang, Z., Hong, W., Zheng, W., Zhou, C., Yin, D., Lin, J., Zou, X., Shao, Z., Yang, H., and Tang, J. (2021). Cogview: Mastering text-to-image generation via transformers.

Doerr, B. and Neumann, F. (2021). A survey on recent progress in the theory of evolutionary algorithms for discrete optimization. 1(4).

Dosovitskiy, A., Beyer, L., Kolesnikov, A., Weissenborn, D., Zhai, X., Unterthiner, T., Dehghani, M., Minderer, M., Heigold, G., Gelly, S., et al. (2020a). An image is worth 16x16 words: Transformers for image recognition at scale.

Dosovitskiy, A., Beyer, L., Kolesnikov, A., Weissenborn, D., Zhai, X., Unterthiner, T., Dehghani, M., Minderer, M., Heigold, G., Gelly, S., Uszkoreit, J., and Houlsby, N. (2020b). An image is worth 16x16 words: Transformers for image recognition at scale.

Dosovitskiy, A., Beyer, L., Kolesnikov, A., Weissenborn, D., Zhai, X., Unterthiner, T., Dehghani, M., Minderer, M., Heigold, G., Gelly, S., Uszkoreit, J., and Houlsby, N. (2020c). An image is worth 16x16 words: Transformers for image recognition at scale. *CoRR*, abs/2010.11929.

Dosovitskiy, A., Beyer, L., Kolesnikov, A., Weissenborn, D., Zhai, X., Unterthiner, T., Dehghani, M., Minderer, M., Heigold, G., Gelly, S., Uszkoreit, J., and Houlsby, N. (2021). An image is worth 16x16 words: Transformers for image recognition at scale. In *9th International Conference on Learning Representations, ICLR 2021, Virtual Event, Austria, May 3-7, 2021*. OpenReview.net.

Dwibedi, D., Aytar, Y., Tompson, J., Sermanet, P., and Zisserman, A. (2021). With a little help from my friends: Nearest-neighbor contrastive learning of visual representations. *CoRR*, abs/2104.14548.

Education, I. C. (2020a). What is Supervised Learning? [Online; accessed 2022-06-29].

Education, I. C. (2020b). What is Unsupervised Learning? [Online; accessed 2022-06-29].

Esser, P., Rombach, R., and Ommer, B. (2020). A note on data biases in generative models.

Esser, P., Rombach, R., and Ommer, B. (2021). Taming transformers for high-resolution image synthesis. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 12873–12883.

Ettinger, A. (2019). What bert is not: Lessons from a new suite of psycholinguistic diagnostics for language models.

Everingham, M., van Gool, L., Williams, C., Winn, J., and Zisserman, A. (2010). The pascal visual object classes (voc) challenge. 88(2):303–338.

Fellbaum, C. (2010). Wordnet. In *Theory and applications of ontology: computer applications*, pages 231–243. Springer.

Fellbaum, C. D. (2000). Wordnet : an electronic lexical database. 76:706.

Fernando, C., Banarse, D., Blundell, C., Zwols, Y., Ha, D., Rusu, A. A., Pritzel,

A., and Wierstra, D. (2017). Pathnet: Evolution channels gradient descent in super neural networks.

Forbes, M., Holtzman, A., and Choi, Y. (2019). Do neural language representations learn physical commonsense?

Gafni, O., Polyak, A., Ashual, O., Sheynin, S., Parikh, D., and Taigman, Y. (2022). Make-a-scene: Scene-based text-to-image generation with human priors.

Galanter, P. (2016). Generative art theory. 1:631.

Gao, J., Li, Z., Nevatia, R., et al. (2017). Knowledge concentration: Learning 100k object classifiers in a single cnn.

Gao, L., Biderman, S., Black, S., Golding, L., Hoppe, T., Foster, C., Phang, J., He, H., Thite, A., Nabeshima, N., et al. (2020). The pile: An 800gb dataset of diverse text for language modeling.

Gatys, L. A., Ecker, A. S., and Bethge, M. (2016). A neural algorithm of artistic style.

Gebru, T., Krause, J., Wang, Y., Chen, D., Deng, J., Aiden, E., and Fei-Fei, L. (2017). Using deep learning and google street view to estimate the demographic makeup of neighborhoods across the united states. 114:201700035.

Gesmundo, A. and Dean, J. (2022). munet: Evolving pretrained deep neural networks into scalable auto-tuning multitask systems.

Gokaslan, A. and Cohen, V. (2019). Openwebtext corpus.

Goodfellow, I., Pouget-Abadie, J., Mirza, M., Xu, B., Warde-Farley, D., Ozair, S., Courville, A., and Bengio, Y. (2014a). Generative adversarial nets. In Ghahramani, Z., Welling, M., Cortes, C., Lawrence, N., and Weinberger, K., editors, *Advances in Neural Information Processing Systems*, volume 27. Curran Associates, Inc.

Goodfellow, I. J., Pouget-Abadie, J., Mirza, M., Xu, B., Warde-Farley, D., Ozair, S., Courville, A., and Bengio, Y. (2014b). Generative adversarial networks.

Goodfellow, I. J., Pouget-Abadie, J., Mirza, M., Xu, B., Warde-Farley, D., Ozair, S., Courville, A., and Bengio, Y. (2014c). Generative adversarial networks.

Goodfellow, I. J., Shlens, J., and Szegedy, C. (2014d). Explaining and harnessing adversarial examples.

Google (2022). Embeddings: Translating to a lower-dimensional space. https://developers.google.com/machine-learning/crash-course/embeddings/translating-to-a-lower-dimensional-space.

Goyal, Y., Khot, T., Summers-Stay, D., Batra, D., and Parikh, D. (2017). 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*, pages 6904–6913.

Grill, J.-B., Strub, F., Altch'e, F., Tallec, C., Richemond, P. H., Buchatskaya, E., Doersch, C., Pires, B. ., Guo, Z., Azar, M. G., Piot, B., Kavukcuoglu, K., Munos, R., and Valko, M. (2020a). Bootstrap your own latent: A new approach to self-supervised learning.

Grill, J.-B., Strub, F., Altché, F., Tallec, C., Richemond, P. H., Buchatskaya, E., Doersch, C., Pires, B. A., Guo, Z. D., Azar, M. G., Piot, B., Kavukcuoglu, K., Munos, R., and Valko, M. (2020b). Bootstrap your own latent: A new approach to self-supervised learning.

Guo, Y., Zhang, L., Hu, Y., He, X., and Gao, J. (2016). Ms-celeb-1m: A dataset and benchmark for large-scale face recognition. In *European conference on computer vision*, pages 87–102. Springer.

Harnad, S. (1990). The symbol grounding problem. 42(1-3):335–346.

Harris, Z. et al. (1954). Distributional hypothesis. 10(23):146–162.

Hart, B. and Risley, T. R. (1995). Meaningful differences in the everyday experience of young american children.

He, K., Chen, X., Xie, S., Li, Y., Dollár, P., and Girshick, R. (2022). Masked autoencoders are scalable vision learners. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 16000–16009.

He, K., Zhang, X., Ren, S., and Sun, J. (2015). Deep residual learning for image recognition.

Henderson, P., Hu, J., Romoff, J., Brunskill, E., Jurafsky, D., and Pineau, J. (2020). Towards the systematic reporting of the energy and carbon footprints of machine learning. 21(248):1–43.

Herdade, S., Kappeler, A., Boakye, K., and Soares, J. (2019). Image captioning: Transforming objects into words. pages 11135–11145.

Heusel, M., Ramsauer, H., Unterthiner, T., Nessler, B., and Hochreiter, S. (2017). Gans trained by a two time-scale update rule converge to a local nash equilibrium. In Guyon, I., Luxburg, U. V., Bengio, S., Wallach, H., Fergus, R., Vishwanathan, S., and Garnett, R., editors, *Advances in Neural Information Processing Systems*, volume 30. Curran Associates, Inc.

Hill, F. and Korhonen, A. (2014). Learning abstract concept embeddings from multi-modal data: Since you probably can't see what i mean. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 255–265.

Hill, F., Reichart, R., and Korhonen, A. (2015). Simlex-999: Evaluating semantic models with (genuine) similarity estimation. 41(4):665–695.

Hinton, G., Vinyals, O., and Dean, J. (2015a). Distilling the knowledge in a neural network.

Hinton, G., Vinyals, O., Dean, J., et al. (2015b). Distilling the knowledge in a neural network. 2(7).

Ho, J., Jain, A., and Abbeel, P. (2020a). Denoising diffusion probabilistic models.

Ho, J., Jain, A., and Abbeel, P. (2020b). Denoising diffusion probabilistic models. In Larochelle, H., Ranzato, M., Hadsell, R., Balcan, M., and Lin, H., editors, *Advances in Neural Information Processing Systems 33: Annual Conference on Neural Information Processing Systems 2020, NeurIPS 2020, December 6-12, 2020, virtual*.

Hochreiter, S. and Schmidhuber, J. (1997). Long short-term memory. 9(8):1735–1780.

Hoffmann, J., Borgeaud, S., Mensch, A., Buchatskaya, E., Cai, T., Rutherford, E., Casas, D. d. L., Hendricks, L. A., Welbl, J., Clark, A., et al. (2022). Training compute-optimal large language models.

Howard, A. G., Zhu, M., Chen, B., Kalenichenko, D., Wang, W., Weyand, T., Andreetto, M., and Adam, H. (2017). Mobilenets: Efficient convolutional neural networks for mobile vision applications. *CoRR*, abs/1704.04861.

Hu, R. and Singh, A. (2021a). Unit: Multimodal multitask learning with a unified transformer. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 1439–1449.

Hu, R. and Singh, A. (2021b). Unit: Multimodal multitask learning with a unified transformer. In *2021 IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 1419–1429.

Huang, L., Wang, W., Chen, J., and Wei, X.-Y. (2019). Attention on attention for image captioning. pages 4633–4642.

Huang, S.-C., Pareek, A., Seyyedi, S., Banerjee, I., and Lungren, M. P. (2020). Fusion of medical imaging and electronic health records using deep learning: a systematic review and implementation guidelines. 3(1):1–9.

Huang, Y., Cheng, Y., Chen, D., Lee, H., Ngiam, J., Le, Q. V., and Chen, Z. (2018). Gpipe: Efficient training of giant neural networks using pipeline parallelism. *CoRR*, abs/1811.06965.

Hudson, D. A. and Manning, C. D. (2019). Gqa: A new dataset for real-world visual reasoning and compositional question answering. In *Proceedings of*

the IEEE/CVF conference on computer vision and pattern recognition, pages 6700–6709.

IV, W. C. S., Kapoor, R., and Ghosh, P. (2021). Multimodal classification: Current landscape, taxonomy and future directions.

Ive, J., Madhyastha, P., and Specia, L. (2019). Distilling translations with visual awareness.

Jacobs, R. A., Jordan, M. I., Nowlan, S. J., and Hinton, G. E. (1991). Adaptive mixtures of local experts. 3(1):79–87.

Jaegle, A., Gimeno, F., Brock, A., Vinyals, O., Zisserman, A., and Carreira, J. (2021a). Perceiver: General perception with iterative attention. In *International conference on machine learning*, pages 4651–4664. PMLR.

Jaegle, A., Gimeno, F., Brock, A., Vinyals, O., Zisserman, A., and Carreira, J. (2021b). Perceiver: General perception with iterative attention. In Meila, M. and Zhang, T., editors, *Proceedings of the 38th International Conference on Machine Learning, ICML 2021, 18-24 July 2021, Virtual Event*, volume 139 of *Proceedings of Machine Learning Research*, pages 4651–4664. PMLR.

Jaspreet (2019). A Concise History of Neural Networks | by Jaspreet | Towards Data Science. [Online; accessed 2022-06-29].

Jean, N., Burke, M., Xie, M., Davis, W. M., Lobell, D. B., and Ermon, S. (2016). Combining satellite imagery and machine learning to predict poverty. 353(6301):790–794.

Jia, C., Yang, Y., Xia, Y., Chen, Y., Parekh, Z., Pham, H., Le, Q. V., Sung, Y., Li, Z., and Duerig, T. (2021a). Scaling up visual and vision-language representation learning with noisy text supervision.

Jia, C., Yang, Y., Xia, Y., Chen, Y.-T., Parekh, Z., Pham, H., Le, Q., Sung, Y.-H., Li, Z., and Duerig, T. (2021b). Scaling up visual and vision-language representation learning with noisy text supervision. In *International Conference on Machine Learning*, pages 4904–4916. PMLR.

Jordan, M. I. and Jacobs, R. A. (1994). Hierarchical mixtures of experts and the em algorithm. 6(2):181–214.

Joseph, K. J., Khan, S. H., Khan, F. S., and Balasubramanian, V. N. (2021). Towards open world object detection. *CoRR*, abs/2103.02603.

Joshi, G., Walambe, R., and Kotecha, K. (2021). A review on explainability in multimodal deep neural nets. 9:59800–59821.

Jumper, J., Evans, R., Pritzel, A., Green, T., Figurnov, M., Ronneberger, O., Tunyasuvunakool, K., Bates, R., Žídek, A., Potapenko, A., Bridgland, A., Meyer, C., Kohl, S. A. A., Ballard, A. J., Cowie, A., Romera-Paredes, B., Nikolov, S., Jain, R., Adler, J., Back, T., Petersen, S., Reiman, D., Clancy, E.,

Zielinski, M., Steinegger, M., Pacholska, M., Berghammer, T., Bodenstein, S., Silver, D., Vinyals, O., Senior, A. W., Kavukcuoglu, K., Kohli, P., and Hassabis, D. (2021). Highly accurate protein structure prediction with AlphaFold. 596(7873):583–589.

Kahatapitiya, K. and Ryoo, M. S. (2021). Swat: Spatial structure within and among tokens.

Kaiser, L., Gomez, A. N., Shazeer, N., Vaswani, A., Parmar, N., Jones, L., and Uszkoreit, J. (2017). One model to learn them all.

Karpathy, A. and Fei-Fei, L. (2014). Deep visual-semantic alignments for generating image descriptions.

Karras, T., Laine, S., and Aila, T. (2019). A style-based generator architecture for generative adversarial networks. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 4401–4410.

Katzman, J. L., Shaham, U., Cloninger, A., Bates, J., Jiang, T., and Kluger, Y. (2018). Deepsurv: personalized treatment recommender system using a cox proportional hazards deep neural network. 18(1):1–12.

Kiela, D. and Bottou, L. (2014). Learning image embeddings using convolutional neural networks for improved multi-modal semantics. In *Proceedings of the 2014 Conference on empirical methods in natural language processing (EMNLP)*, pages 36–45.

Kiela, D., Conneau, A., Jabri, A., and Nickel, M. (2017). Learning visually grounded sentence representations.

Kingma, D. P. and Welling, M. (2013). Auto-encoding variational bayes.

Kingma, D. P. and Welling, M. (2019). An introduction to variational autoencoders.

Kiros, J., Chan, W., and Hinton, G. (2018). Illustrative language understanding: Large-scale visual grounding with image search. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 922–933.

Koehn, P. (2005). Europarl: A parallel corpus for statistical machine translation. In *Proceedings of machine translation summit x: papers*, pages 79–86.

Kolesnikov, A., Beyer, L., Zhai, X., Puigcerver, J., Yung, J., Gelly, S., and Houlsby, N. (2019). Large scale learning of general visual representations for transfer. 2(8).

Kopper, P., Wiegrebe, S., Bischl, B., Bender, A., and Rügamer, D. (2022). Deeppamm: Deep piecewise exponential additive mixed models for complex hazard structures in survival analysis. In *Pacific-Asia Conference on Knowledge Discovery and Data Mining*, pages 249–261. Springer.

Kottur, S., Vedantam, R., Moura, J. M., and Parikh, D. (2016). Visual word2vec (vis-w2v): Learning visually grounded word embeddings using abstract scenes. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 4985–4994.

Krishna, R., Zhu, Y., Groth, O., Johnson, J., Hata, K., Kravitz, J., Chen, S., Kalantidis, Y., Li, L.-J., Shamma, D. A., Bernstein, M., and Fei-Fei, L. (2016). Visual genome: Connecting language and vision using crowdsourced dense image annotations.

Krishna, R., Zhu, Y., Groth, O., Johnson, J., Hata, K., Kravitz, J., Chen, S., Kalantidis, Y., Li, L.-J., Shamma, D. A., et al. (2017). Visual genome: Connecting language and vision using crowdsourced dense image annotations. 123(1):32–73.

Krizhevsky, A., Hinton, G., et al. (2009). Learning multiple layers of features from tiny images.

Krizhevsky, A., Sutskever, I., and Hinton, G. E. (2012a). Imagenet classification with deep convolutional neural networks. In Pereira, F., Burges, C., Bottou, L., and Weinberger, K., editors, *Advances in Neural Information Processing Systems*, volume 25. Curran Associates, Inc.

Krizhevsky, A., Sutskever, I., and Hinton, G. E. (2012b). Imagenet classification with deep convolutional neural networks. 25.

Kudo, T. and Richardson, J. (2018). SentencePiece: A simple and language independent subword tokenizer and detokenizer for neural text processing. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 66–71. Association for Computational Linguistics.

Kuznetsova, A., Rom, H., Alldrin, N., Uijlings, J., Krasin, I., Pont-Tuset, J., Kamali, S., Popov, S., Malloci, M., Kolesnikov, A., et al. (2020). The open images dataset v4. 128(7):1956–1981.

Kynkäänniemi, T., Karras, T., Laine, S., Lehtinen, J., and Aila, T. (2019). Improved precision and recall metric for assessing generative models.

Law, S., Paige, B., and Russell, C. (2019). Take a look around. 10(5):1–19.

Lazaridou, A., Pham, N. T., and Baroni, M. (2015). Combining language and vision with a multimodal skip-gram model.

LeCun, Y. (2022). A path towards autonomous machine intelligence version 0.9. 2, 2022-06-27.

Lewis, M., Liu, Y., Goyal, N., Ghazvininejad, M., Mohamed, A., Levy, O., Stoyanov, V., and Zettlemoyer, L. (2020). BART: Denoising sequence-to-sequence pre-training for natural language generation, translation, and comprehension. In *Proceedings of the 58th Annual Meeting of the Association for*

Computational Linguistics, pages 7871–7880. Association for Computational Linguistics.

Lewkowycz, A., Andreassen, A., Dohan, D. M., Dyer, E. S., Michalewski, H., Ramasesh, V., Slone, A., Anil, C., Schlag, I., Gutman-Solo, T., Wu, Y., Neyshabur, B., Gur-Ari, G., and Misra, V. (2022). Solving quantitative reasoning problems with language models.

Lialin, V., Zhao, K., Shivagunde, N., and Rumshisky, A. (2022). Life after bert: What do other muppets understand about language?

Lin, C.-Y. (2004). ROUGE: A package for automatic evaluation of summaries. In *Text Summarization Branches Out*, pages 74–81. Association for Computational Linguistics.

Lin, T.-Y., Maire, M., Belongie, S., Bourdev, L., Girshick, R., Hays, J., Perona, P., Ramanan, D., Zitnick, C. L., and Dollár, P. (2014a). Microsoft coco: Common objects in context.

Lin, T.-Y., Maire, M., Belongie, S., Hays, J., Perona, P., Ramanan, D., Dollár, P., and Zitnick, C. L. (2014b). Microsoft coco: Common objects in context. In *European conference on computer vision*, pages 740–755. Springer.

Lin, T.-Y., Maire, M., Belongie, S., Hays, J., Perona, P., Ramanan, D., Dollár, P., and Zitnick, C. L. (2014c). Microsoft coco: Common objects in context. In *Computer Vision – ECCV 2014*, pages 740–755. Springer International Publishing.

Lin, Y., Tan, Y. C., and Frank, R. (2019). Open sesame: Getting inside BERT's linguistic knowledge. In *Proceedings of the 2019 ACL Workshop BlackboxNLP: Analyzing and Interpreting Neural Networks for NLP*. Association for Computational Linguistics.

Liu, N. F., Gardner, M., Belinkov, Y., Peters, M. E., and Smith, N. A. (2019a). Linguistic knowledge and transferability of contextual representations.

Liu, Y., Ott, M., Goyal, N., Du, J., Joshi, M., Chen, D., Levy, O., Lewis, M., Zettlemoyer, L., and Stoyanov, V. (2019b). Roberta: A robustly optimized bert pretraining approach.

Lottick, K., Susai, S., Friedler, S. A., and Wilson, J. P. (2019). Energy usage reports: Environmental awareness as part of algorithmic accountability.

Lu, J., Batra, D., Parikh, D., and Lee, S. (2019a). Vilbert: Pretraining task-agnostic visiolinguistic representations for vision-and-language tasks.

Lu, J., Batra, D., Parikh, D., and Lee, S. (2019b). Vilbert: Pretraining task-agnostic visiolinguistic representations for vision-and-language tasks. 32.

Lu, Y., Zhu, W., Wang, X. E., Eckstein, M., and Wang, W. Y. (2022). Imagination-augmented natural language understanding.

Mahajan, D., Girshick, R., Ramanathan, V., He, K., Paluri, M., Li, Y., Bharambe, A., and Van Der Maaten, L. (2018). Exploring the limits of weakly supervised pretraining. In *Proceedings of the European conference on computer vision (ECCV)*, pages 181–196.

Manning, C., Goldie, A., and Hewitt, J. (2022). Stanford cs224n: Natural language processing with deep learning. https://web.stanford.edu/class/cs224n/slides/.

Mayer, T. and Cysouw, M. (2014). Creating a massively parallel bible corpus. 135(273):40.

Mccormack, J. and Gambardella, C. C. (2022). Growing and evolving 3-d prints. 26(1):88–99.

MICHAEL BARTHEL, GALEN STOCKING, J. H. and MITCHELL, A. (2016). Reddit news users more likely to be male, young and digital in their news preferences.

Midjourney (2022). Midjourney. https://www.midjourney.com/. Accessed: 2022-09-12.

Mikolov, T., Chen, K., Corrado, G., and Dean, J. (2013a). Efficient estimation of word representations in vector space.

Mikolov, T., Chen, K., Corrado, G., and Dean, J. (2013b). Efficient estimation of word representations in vector space.

Mikolov, T., Le, Q. V., and Sutskever, I. (2013c). Exploiting similarities among languages for machine translation.

Mikolov, T., Sutskever, I., Chen, K., Corrado, G., and Dean, J. (2013d). Distributed representations of words and phrases and their compositionality.

Mineault, P. (2021). Unsupervised models of the brain.

Mircosoft (2019). Evaluate:detection.

Mishkin, P., Ahmad, L., Brundage, M., Krueger, G., and Sastry, G. (2022). Dall · e 2 preview - risks and limitations.

Mordvintsev, A. (2015). Inceptionism: Going deeper into neural networks.

Mustafa, B., Riquelme, C., Puigcerver, J., Jenatton, R., and Houlsby, N. (2022). Multimodal contrastive learning with limoe: the language-image mixture of experts.

Nagrani, A., Yang, S., Arnab, A., Jansen, A., Schmid, C., and Sun, C. (2021). Attention bottlenecks for multimodal fusion. In Ranzato, M., Beygelzimer, A., Dauphin, Y. N., Liang, P., and Vaughan, J. W., editors, *Advances in Neural Information Processing Systems 34: Annual Conference on Neural*

Information Processing Systems 2021, NeurIPS 2021, December 6-14, 2021, virtual, pages 14200–14213.

Nichol, A., Dhariwal, P., Ramesh, A., Shyam, P., Mishkin, P., McGrew, B., Sutskever, I., and Chen, M. (2021a). Glide: Towards photorealistic image generation and editing with text-guided diffusion models.

Nichol, A., Dhariwal, P., Ramesh, A., Shyam, P., Mishkin, P., McGrew, B., Sutskever, I., and Chen, M. (2021b). GLIDE: towards photorealistic image generation and editing with text-guided diffusion models.

OpenAI (2021). Dall-e.

Papineni, K., Roukos, S., Ward, T., and Zhu, W.-J. (2002). Bleu: a method for automatic evaluation of machine translation. In *Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics*, pages 311–318. Association for Computational Linguistics.

Parcalabescu, L., Cafagna, M., Muradjan, L., Frank, A., Calixto, I., and Gatt, A. (2022). VALSE: A task-independent benchmark for vision and language models centered on linguistic phenomena. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 8253–8280. Association for Computational Linguistics.

Patashnik, O., Wu, Z., Shechtman, E., Cohen-Or, D., and Lischinski, D. (2021). Styleclip: Text-driven manipulation of stylegan imagery.

Pennington, J., Socher, R., and Manning, C. D. (2014). Glove: Global vectors for word representation. In *Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP)*, pages 1532–1543.

Perez, E., Kiela, D., and Cho, K. (2021a). True few-shot learning with language models.

Perez, E., Kiela, D., and Cho, K. (2021b). True few-shot learning with language models. 34:11054–11070.

Pezzelle, S., Takmaz, E., and Fernández, R. (2021). Word representation learning in multimodal pre-trained transformers: An intrinsic evaluation. 9:1563–1579.

Pilehvar, M. T. and Camacho-Collados, J. (2021). *Embeddings in Natural Language Processing*. Springer International Publishing.

Pont-Tuset, J., Uijlings, J., Changpinyo, S., Soricut, R., and Ferrari, V. (2020). Connecting vision and language with localized narratives. In *European conference on computer vision*, pages 647–664. Springer.

Prabhu, V. U. and Birhane, A. (2020). Large image datasets: A pyrrhic win for computer vision?

Pölsterl, S., Sarasua, I., Gutiérrez-Becker, B., and Wachinger, C. (2019). A wide

and deep neural network for survival analysis from anatomical shape and tabular clinical data. In *Joint European Conference on Machine Learning and Knowledge Discovery in Databases*, pages 453–464. Springer.

Qiao, H., Liu, V., and Chilton, L. (2022). Initial images: Using image prompts to improve subject representation in multimodal ai generated art. In *Creativity and Cognition*, pages 15–28.

Qiao, T., Zhang, J., Xu, D., and Tao, D. (2019). Mirrorgan: Learning text-to-image generation by redescription.

R Core Team (2018). *R: A Language and Environment for Statistical Computing*. R Foundation for Statistical Computing.

Radford, A., Kim, J. W., Hallacy, C., Ramesh, A., Goh, G., Agarwal, S., Sastry, G., Askell, A., Mishkin, P., Clark, J., et al. (2021a). Learning transferable visual models from natural language supervision. In *International Conference on Machine Learning*, pages 8748–8763. PMLR.

Radford, A., Kim, J. W., Hallacy, C., Ramesh, A., Goh, G., Agarwal, S., Sastry, G., Askell, A., Mishkin, P., Clark, J., Krueger, G., and Sutskever, I. (2021b). Learning transferable visual models from natural language supervision.

Radford, A., Kim, J. W., Hallacy, C., Ramesh, A., Goh, G., Agarwal, S., Sastry, G., Askell, A., Mishkin, P., Clark, J., Krueger, G., and Sutskever, I. (2021c). Learning transferable visual models from natural language supervision. In Meila, M. and Zhang, T., editors, *Proceedings of the 38th International Conference on Machine Learning, ICML 2021, 18-24 July 2021, Virtual Event*, volume 139 of *Proceedings of Machine Learning Research*, pages 8748–8763. PMLR.

Radford, A., Narasimhan, K., Salimans, T., and Sutskever, I. (2018). Improving language understanding by generative pre-training.

Radford, A., Wu, J., Child, R., Luan, D., Amodei, D., and Sutskever, I. (2019a). Language models are unsupervised multitask learners.

Radford, A., Wu, J., Child, R., Luan, D., Amodei, D., Sutskever, I., et al. (2019b). Language models are unsupervised multitask learners. 1(8):9.

Raffel, C., Shazeer, N., Roberts, A., Lee, K., Narang, S., Matena, M., Zhou, Y., Li, W., and Liu, P. J. (2019a). Exploring the limits of transfer learning with a unified text-to-text transformer.

Raffel, C., Shazeer, N., Roberts, A., Lee, K., Narang, S., Matena, M., Zhou, Y., Li, W., and Liu, P. J. (2019b). Exploring the limits of transfer learning with a unified text-to-text transformer.

Raghu, M., Poole, B., Kleinberg, J., Ganguli, S., and Sohl-Dickstein, J. (2016). On the expressive power of deep neural networks.

Rajpurkar, P., Jia, R., and Liang, P. (2018). Know what you don't know: Unanswerable questions for squad.

Rajpurkar, P., Zhang, J., Lopyrev, K., and Liang, P. (2016). Squad: 100,000+ questions for machine comprehension of text.

Ramachandran, P., Parmar, N., Vaswani, A., Bello, I., Levskaya, A., and Shlens, J. (2019). Stand-alone self-attention in vision models. *CoRR*, abs/1906.05909.

Ramesh, A., Dhariwal, P., Nichol, A., Chu, C., and Chen, M. (2022a). Hierarchical text-conditional image generation with clip latents.

Ramesh, A., Dhariwal, P., Nichol, A., Chu, C., and Chen, M. (2022b). Hierarchical text-conditional image generation with clip latents. 2022.

Ramesh, A., Pavlov, M., Goh, G., Gray, S., Voss, C., Radford, A., Chen, M., and Sutskever, I. (2021a). Zero-shot text-to-image generation.

Ramesh, A., Pavlov, M., Goh, G., Gray, S., Voss, C., Radford, A., Chen, M., and Sutskever, I. (2021b). Zero-shot text-to-image generation.

Ramesh, A., Pavlov, M., Goh, G., Gray, S., Voss, C., Radford, A., Chen, M., and Sutskever, I. (2021c). Zero-shot text-to-image generation. In Meila, M. and Zhang, T., editors, *Proceedings of the 38th International Conference on Machine Learning, ICML 2021, 18-24 July 2021, Virtual Event*, volume 139 of *Proceedings of Machine Learning Research*, pages 8821–8831. PMLR.

Ramesh, A., Pavlov, M., Goh, G., Gray, S., Voss, C., Radford, A., Chen, M., and Sutskever, I. (2021d). Zero-shot text-to-image generation. In Meila, M. and Zhang, T., editors, *Proceedings of the 38th International Conference on Machine Learning*, volume 139 of *Proceedings of Machine Learning Research*, pages 8821–8831. PMLR.

Rebuffi, S.-A., Bilen, H., and Vedaldi, A. (2017). Learning multiple visual domains with residual adapters. In Guyon, I., Luxburg, U. V., Bengio, S., Wallach, H., Fergus, R., Vishwanathan, S., and Garnett, R., editors, *Advances in Neural Information Processing Systems*, volume 30. Curran Associates, Inc.

Recht, B., Roelofs, R., Schmidt, L., and Shankar, V. (2019). Do imagenet classifiers generalize to imagenet? In *International Conference on Machine Learning*, pages 5389–5400. PMLR.

Reed, S., Zolna, K., Parisotto, E., Colmenarejo, S. G., Novikov, A., Barth-Maron, G., Gimenez, M., Sulsky, Y., Kay, J., Springenberg, J. T., Eccles, T., Bruce, J., Razavi, A., Edwards, A., Heess, N., Chen, Y., Hadsell, R., Vinyals, O., Bordbar, M., and de Freitas, N. (2022). A generalist agent.

Reed, S. E., Akata, Z., Mohan, S., Tenka, S., Schiele, B., and Lee, H. (2016a). Learning what and where to draw.

Reed, S. E., Akata, Z., Schiele, B., and Lee, H. (2016b). Learning deep representations of fine-grained visual descriptions.

Reed, S. E., Akata, Z., Yan, X., Logeswaran, L., Schiele, B., and Lee, H. (2016c). Generative adversarial text to image synthesis.

Ren, S., He, K., Girshick, R., and Sun, J. (2015). Faster r-cnn: Towards real-time object detection with region proposal networks. 28.

Rennie, S. J., Marcheret, E., Mroueh, Y., Ross, J., and Goel, V. (2017). Self-critical sequence training for image captioning. In *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1179–1195.

Ribeiro, M. T., Wu, T., Guestrin, C., and Singh, S. (2020). Beyond accuracy: Behavioral testing of nlp models with checklist.

Riquelme, C., Puigcerver, J., Mustafa, B., Neumann, M., Jenatton, R., Susano Pinto, A., Keysers, D., and Houlsby, N. (2021). Scaling vision with sparse mixture of experts. In Ranzato, M., Beygelzimer, A., Dauphin, Y., Liang, P., and Vaughan, J. W., editors, *Advances in Neural Information Processing Systems*, volume 34, pages 8583–8595. Curran Associates, Inc.

Ritchie, H., Roser, M., and Rosado, P. (2020). co_2 and greenhouse gas emissions. https://ourworldindata.org/co2-and-other-greenhouse-gas-emissions.

Rombach, R., Blattmann, A., Lorenz, D., Esser, P., and Ommer, B. (2021). High-resolution image synthesis with latent diffusion models.

Rombach, R., Blattmann, A., Lorenz, D., Esser, P., and Ommer, B. (2022). Stablediffusion. https://github.com/CompVis/stable-diffusion. Accessed: 2022-09-12.

Rosset, C. (2020). Turing-nlg: A 17-billion-parameter language model by microsoft. 1(2).

Russakovsky, O., Deng, J., Su, H., Krause, J., Satheesh, S., Ma, S., Huang, Z., Karpathy, A., Khosla, A., Bernstein, M., Berg, A. C., and Fei-Fei, L. (2015). Imagenet large scale visual recognition challenge. 115(3):211–252.

Rügamer, D., Kolb, C., and Klein, N. (2020). Semi-structured deep distributional regression: Combining structured additive models and deep learning.

Saharia, C., Chan, W., Saxena, S., Li, L., Whang, J., Denton, E., Ghasemipour, S. K. S., Ayan, B. K., Mahdavi, S. S., Lopes, R. G., Salimans, T., Ho, J., Fleet, D. J., and Norouzi, M. (2022a). Photorealistic text-to-image diffusion models with deep language understanding.

Saharia, C., Chan, W., Saxena, S., Li, L., Whang, J., Denton, E., Ghasemipour, S. K. S., Ayan, B. K., Mahdavi, S. S., Lopes, R. G., Salimans, T., Ho, J., Fleet, D. J., and Norouzi, M. (2022b). Photorealistic text-to-image diffusion models with deep language understanding.

Saifee, M. (2020). Gpt-3: The new mighty language model from openai. https://towardsdatascience.com/gpt-3-the-new-mighty-language-model-from-openai-a74ff35346fc.

Sajjadi, M. S. M., Bachem, O., Lucic, M., Bousquet, O., and Gelly, S. (2018). Assessing generative models via precision and recall.

Salimans, T., Goodfellow, I. J., Zaremba, W., Cheung, V., Radford, A., and Chen, X. (2016). Improved techniques for training gans.

Schick, T. and Schütze, H. (2020). Exploiting cloze questions for few shot text classification and natural language inference.

Schuhmann, C. (2022). Laion-400-million open dataset.

Schuhmann, C., Vencu, R., Beaumont, R., Kaczmarczyk, R., Mullis, C., Katta, A., Coombes, T., Jitsev, J., and Komatsuzaki, A. (2021a). Laion-400m: Open dataset of clip-filtered 400 million image-text pairs.

Schuhmann, C., Vencu, R., Beaumont, R., Kaczmarczyk, R., Mullis, C., Katta, A., Coombes, T., Jitsev, J., and Komatsuzaki, A. (2021b). LAION-400M: open dataset of clip-filtered 400 million image-text pairs.

Sejnowski, T. J. (2020). The unreasonable effectiveness of deep learning in artificial intelligence.

Sennrich, R., Haddow, B., and Birch, A. (2015a). Neural machine translation of rare words with subword units.

Sennrich, R., Haddow, B., and Birch, A. (2015b). Neural machine translation of rare words with subword units.

Sennrich, R., Haddow, B., and Birch, A. (2016). Neural machine translation of rare words with subword units. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1715–1725. Association for Computational Linguistics.

Shah, D. (2022). Self-Supervised Learning and Its Applications - neptune.ai. [Online; accessed 2022-06-29].

Shao, S., Li, Z., Zhang, T., Peng, C., Yu, G., Zhang, X., Li, J., and Sun, J. (2019). Objects365: A large-scale, high-quality dataset for object detection. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 8430–8439.

Shazeer, N., Mirhoseini, A., Maziarz, K., Davis, A., Le, Q., Hinton, G., and Dean, J. (2017). Outrageously large neural networks: The sparsely-gated mixture-of-experts layer.

Shekhar, R., Pezzelle, S., Klimovich, Y., Herbelot, A., Nabi, M., Sangineto, E., and Bernardi, R. (2017). Foil it! find one mismatch between image and language caption.

Shen, S., Li, L. H., Tan, H., Bansal, M., Rohrbach, A., Chang, K.-W., Yao, Z., and Keutzer, K. (2021). How much can clip benefit vision-and-language tasks?

Sheng, E., Chang, K.-W., Natarajan, P., and Peng, N. (2019). The woman worked as a babysitter: On biases in language generation.

Shonenkov, A. (2021). rudall-e.

Shvetsova, N., Chen, B., Rouditchenko, A., Thomas, S., Kingsbury, B., Feris, R., Harwath, D., Glass, J., and Kuehne, H. (2021). Everything at once - multi-modal fusion transformer for video retrieval.

Sikarwar, A. and Kreiman, G. (2022). On the efficacy of co-attention transformer layers in visual question answering.

Silberer, C. and Lapata, M. (2014). Learning grounded meaning representations with autoencoders. In *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 721–732.

Simonyan, K. and Zisserman, A. (2014). Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*.

Singh, A., Hu, R., Goswami, V., Couairon, G., Galuba, W., Rohrbach, M., and Kiela, D. (2022). Flava: A foundational language and vision alignment model. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 15638–15650.

Sirko, W., Kashubin, S., Ritter, M., Annkah, A., Bouchareb, Y. S. E., Dauphin, Y. N., Keysers, D., Neumann, M., Cissé, M., and Quinn, J. (2021). Continental-scale building detection from high resolution satellite imagery.

Snell, C. (2021). Understanding vq-vae. https://ml.berkeley.edu/blog/posts/vq-vae/. Accessed: 2022-09-12.

Socher, R. and Fei-fei, L. (2010). Connecting modalities: Semi-supervised segmentation and annotation of images using unaligned text corpora. In *In IEEE Computer Society Conference on Computer Vision and Pattern Recognition*.

Soderlund, J. and Blair, A. (2018). Adversarial image generation using evolution and deep learning. In *2018 IEEE Congress on Evolutionary Computation (CEC)*, pages 1–8.

Sohl-Dickstein, J., Weiss, E. A., Maheswaranathan, N., and Ganguli, S. (2015). Deep unsupervised learning using nonequilibrium thermodynamics.

Srinivasan, K., Raman, K., Chen, J., Bendersky, M., and Najork, M. (2021). Wit: Wikipedia-based image text dataset for multimodal multilingual machine learning. In *Proceedings of the 44th International ACM SIGIR Con-*

ference on Research and Development in Information Retrieval, pages 2443–2449.

Srinivasan, R. and Uchino, K. (2021). Biases in generative art: A causal look from the lens of art history. In *Proceedings of the 2021 ACM Conference on Fairness, Accountability, and Transparency*, pages 41–51.

Srivastava, A., Rastogi, A., Rao, A., Shoeb, A. A. M., Abid, A., Fisch, A., Brown, A. R., Santoro, A., Gupta, A., Garriga-Alonso, A., et al. (2022). Beyond the imitation game: Quantifying and extrapolating the capabilities of language models.

Steiner, A., Kolesnikov, A., Zhai, X., Wightman, R., Uszkoreit, J., and Beyer, L. (2021). How to train your vit? data, augmentation, and regularization in vision transformers.

Strubell, E., Ganesh, A., and McCallum, A. (2019a). Energy and policy considerations for deep learning in NLP. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 3645–3650. Association for Computational Linguistics.

Strubell, E., Ganesh, A., and McCallum, A. (2019b). Energy and policy considerations for deep learning in nlp.

Strubell, E., Ganesh, A., and McCallum, A. (2019c). Energy and policy considerations for deep learning in nlp.

Sulubacak, U., Caglayan, O., Grönroos, S., Rouhe, A., Elliott, D., Specia, L., and Tiedemann, J. (2020). Multimodal machine translation through visuals and speech. 34(2-3):97–147.

Sun, C., Shrivastava, A., Singh, S., and Gupta, A. (2017). Revisiting unreasonable effectiveness of data in deep learning era. In *Proceedings of the IEEE international conference on computer vision*, pages 843–852.

Sun, Q., Wang, Y., Xu, C., Zheng, K., Yang, Y., Hu, H., Xu, F., Zhang, J., Geng, X., and Jiang, D. (2021). Multimodal dialogue response generation.

Sutskever, I., Vinyals, O., and Le, Q. V. (2014). Sequence to sequence learning with neural networks.

Sutton, R. S. (2019). The bitter lesson.

Szegedy, C., Vanhoucke, V., Ioffe, S., Shlens, J., and Wojna, Z. (2015). Rethinking the inception architecture for computer vision.

Tan, H. and Bansal, M. (2020). Vokenization: Improving language understanding with contextualized, visual-grounded supervision.

Tan, M. and Le, Q. V. (2019a). Efficientnet: Rethinking model scaling for convolutional neural networks.

Tan, M. and Le, Q. V. (2019b). Efficientnet: Rethinking model scaling for convolutional neural networks. *CoRR*, abs/1905.11946.

Tao, M., Tang, H., Wu, S., Sebe, N., Wu, F., and Jing, X. (2020). DF-GAN: deep fusion generative adversarial networks for text-to-image synthesis.

techslang (2020). What is Self-Supervised Learning? — Definition by Techslang. [Online; accessed 2022-06-29].

Theis, L., Oord, A. v. d., and Bethge, M. (2015). A note on the evaluation of generative models.

Tian, Y., Krishnan, D., and Isola, P. (2020). Contrastive multiview coding. In *European conference on computer vision*, pages 776–794. Springer.

Tiu, E. (2021). Understanding Contrastive Learning | by Ekin Tiu | Towards Data Science. [Online; accessed 2022-06-29].

Tong, C., Li, J., Lang, C., Kong, F., Niu, J., and Rodrigues, J. J. (2018). An efficient deep model for day-ahead electricity load forecasting with stacked denoising auto-encoders. 117:267–273.

Torralba, A. and Efros, A. A. (2011). Unbiased look at dataset bias. In *CVPR 2011*, pages 1521–1528. IEEE.

Uppal, S., Bhagat, S., Hazarika, D., Majumder, N., Poria, S., Zimmermann, R., and Zadeh, A. (2022). Multimodal research in vision and language: A review of current and emerging trends. 77:149–171.

Vale-Silva, L. A. and Rohr, K. (2021). Long-term cancer survival prediction using multimodal deep learning. 11(1):1–12.

van den Oord, A., Vinyals, O., and Kavukcuoglu, K. (2017). Neural discrete representation learning.

Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, L., and Polosukhin, I. (2017a). Attention is all you need. In Guyon, I., Luxburg, U. V., Bengio, S., Wallach, H., Fergus, R., Vishwanathan, S., and Garnett, R., editors, *Advances in Neural Information Processing Systems*, volume 30. Curran Associates, Inc.

Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, L., and Polosukhin, I. (2017b). Attention is all you need. 30.

Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, L., and Polosukhin, I. (2017c). Attention is all you need.

Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, L., and Polosukhin, I. (2017d). Attention is all you need. In Guyon, I., Luxburg, U. V., Bengio, S., Wallach, H., Fergus, R., Vishwanathan, S., and Garnett, R., editors, *Advances in Neural Information Processing Systems*, volume 30. Curran Associates, Inc.

Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, L., and Polosukhin, I. (2017e). Attention is all you need.

Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, L., and Polosukhin, I. (2017f). Attention is all you need. In Guyon, I., von Luxburg, U., Bengio, S., Wallach, H. M., Fergus, R., Vishwanathan, S. V. N., and Garnett, R., editors, *Advances in Neural Information Processing Systems 30: Annual Conference on Neural Information Processing Systems 2017, December 4-9, 2017, Long Beach, CA, USA*, pages 5998–6008.

Vedantam, R., Zitnick, C. L., and Parikh, D. (2015). Cider: Consensus-based image description evaluation. In *2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 4566–4575.

Vinyals, O., Toshev, A., Bengio, S., and Erhan, D. (2015). Show and tell: A neural image caption generator. pages 3156–3164.

Voita, E., Talbot, D., Moiseev, F., Sennrich, R., and Titov, I. (2019). Analyzing multi-head self-attention: Specialized heads do the heavy lifting, the rest can be pruned.

Wang, A., Singh, A., Michael, J., Hill, F., Levy, O., and Bowman, S. R. (2018). Glue: A multi-task benchmark and analysis platform for natural language understanding.

Wang, J. and Li, S. (2018). Detection and classification of acoustic scenes and events 2018 self-attention mechanism based system for dcase2018 challenge task1 and task4.

Wang, P., Yang, A., Men, R., Lin, J., Bai, S., Li, Z., Ma, J., Zhou, C., Zhou, J., and Yang, H. (2022). OFA: Unifying architectures, tasks, and modalities through a simple sequence-to-sequence learning framework. In Chaudhuri, K., Jegelka, S., Song, L., Szepesvari, C., Niu, G., and Sabato, S., editors, *Proceedings of the 39th International Conference on Machine Learning*, volume 162 of *Proceedings of Machine Learning Research*, pages 23318–23340. PMLR.

Wang, Q., Li, R., Wang, Q., and Chen, S. (2021). Non-fungible token (nft): Overview, evaluation, opportunities and challenges.

Website (2020). Localized narratives data and visualization.

Wei, C., Fan, H., Xie, S., Wu, C.-Y., Yuille, A., and Feichtenhofer, C. (2022). Masked feature prediction for self-supervised visual pre-training. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 14668–14678.

Weng, L. (2018). From autoencoder to beta-vae.

Weng, L. (2021). What are diffusion models?

Wenzek, G., Lachaux, M.-A., Conneau, A., Chaudhary, V., Guzmán, F., Joulin, A., and Grave, E. (2019). Ccnet: Extracting high quality monolingual datasets from web crawl data.

Wu, C., Liang, J., Ji, L., Yang, F., Fang, Y., Jiang, D., and Duan, N. (2021). NÜwa: Visual synthesis pre-training for neural visual world creation.

WZRD (2020). Wzrd.

Xiao, J., Hays, J., Ehinger, K. A., Oliva, A., and Torralba, A. (2010). Sun database: Large-scale scene recognition from abbey to zoo. In *2010 IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pages 3485–3492.

Xie, Q., Hovy, E. H., Luong, M., and Le, Q. V. (2019). Self-training with noisy student improves imagenet classification. *CoRR*, abs/1911.04252.

Xu, K., Ba, J., Kiros, R., Cho, K., Courville, A., Salakhutdinov, R., Zemel, R., and Bengio, Y. (2015). Show, attend and tell: Neural image caption generation with visual attention.

Xu, T., Zhang, P., Huang, Q., Zhang, H., Gan, Z., Huang, X., and He, X. (2017). Attngan: Fine-grained text to image generation with attentional generative adversarial networks.

Xue, L., Constant, N., Roberts, A., Kale, M., Al-Rfou, R., Siddhant, A., Barua, A., and Raffel, C. (2020). mt5: A massively multilingual pre-trained text-to-text transformer.

Yang, X., Tang, K., Zhang, H., and Cai, J. (2019). Auto-encoding scene graphs for image captioning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*.

Yann, L. and Ishan, M. (2021). Self-supervised learning: The dark matter of intelligence.

Yao, B. Z., Yang, X., Lin, L., Lee, M. W., and Zhu, S.-C. (2010). I2t: Image parsing to text description. 98(8):1485–1508.

Yao, J., Zhu, X., Zhu, F., and Huang, J. (2017). Deep correlational learning for survival prediction from multi-modality data. In *Medical Image Computing and Computer-Assisted Intervention - MICCAI 2017*, pages 406–414. Springer International Publishing.

Yao, T., Pan, Y., Li, Y., and Mei, T. (2018a). Exploring visual relationship for image captioning.

Yao, T., Pan, Y., Li, Y., and Mei, T. (2018b). Exploring visual relationship for image captioning.

You, J., Li, X., Low, M., Lobell, D., and Ermon, S. (2017). Deep gaussian process for crop yield prediction based on remote sensing data. In *Proceedings*

of the Thirty-First AAAI Conference on Artificial Intelligence, AAAI'17, pages 4559–4565. AAAI Press.

Young, P., Lai, A., Hodosh, M., and Hockenmaier, J. (2014). From image descriptions to visual denotations: New similarity metrics for semantic inference over event descriptions. 2:67–78.

Yu, J., Li, X., Koh, J. Y., Zhang, H., Pang, R., Qin, J., Ku, A., Xu, Y., Baldridge, J., and Wu, Y. (2021). Vector-quantized image modeling with improved VQGAN.

Yu, J., Xu, Y., Koh, J., Luong, T., Baid, G., Vasudevan, V., Ku, A., Yang, Y., Ayan, B., Hutchinson, B., Han, W., Parekh, Z., Li, X., Zhang, H., Baldridge, J., and Wu, Y. (2022a). Scaling autoregressive models for content-rich text-to-image generation.

Yu, J., Xu, Y., Koh, J. Y., Luong, T., Baid, G., Wang, Z., Vasudevan, V., Ku, A., Yang, Y., Ayan, B. K., Hutchinson, B., Han, W., Parekh, Z., Li, X., Zhang, H., Baldridge, J., and Wu, Y. (2022b). Scaling autoregressive models for content-rich text-to-image generation.

Yuan, L., Chen, D., Chen, Y.-L., Codella, N., Dai, X., Gao, J., Hu, H., Huang, X., Li, B., Li, C., et al. (2021). Florence: A new foundation model for computer vision.

Yuan, S., Shuai, Z., Jiahong, L., Zhao, X., Hanyu, Z., and Jie, T. (2022). Wudaomm: A large-scale multi-modal dataset for pre-training models.

Zagoruyko, S. and Komodakis, N. (2016). Wide residual networks. *CoRR*, abs/1605.07146.

Zellers, R., Bisk, Y., Farhadi, A., and Choi, Y. (2019). From recognition to cognition: Visual commonsense reasoning. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 6720–6731.

Zellers, R., Bisk, Y., Schwartz, R., and Choi, Y. (2018). Swag: A large-scale adversarial dataset for grounded commonsense inference.

Zeng, A., Wong, A., Welker, S., Choromanski, K., Tombari, F., Purohit, A., Ryoo, M., Sindhwani, V., Lee, J., Vanhoucke, V., et al. (2022). Socratic models: Composing zero-shot multimodal reasoning with language.

Zhang, C., Yang, Z., He, X., and Deng, L. (2020a). Multimodal intelligence: Representation learning, information fusion, and applications. 14(3):478–493.

Zhang, H., Xu, T., Li, H., Zhang, S., Huang, X., Wang, X., and Metaxas, D. N. (2016a). Stackgan: Text to photo-realistic image synthesis with stacked generative adversarial networks.

Zhang, P., Goyal, Y., Summers-Stay, D., Batra, D., and Parikh, D. (2016b). Yin and yang: Balancing and answering binary visual questions. In *Proceedings*

of the IEEE conference on computer vision and pattern recognition, pages 5014–5022.

Zhang, Y., Jiang, H., Miura, Y., Manning, C. D., and Langlotz, C. P. (2020b). Contrastive learning of medical visual representations from paired images and text.

Zhou, B., Zhao, H., Puig, X., Fidler, S., Barriuso, A., and Torralba, A. (2017). Scene parsing through ade20k dataset. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 633–641.

Zhou, Y., Roy, S., Abdolrashidi, A., Wong, D., Ma, P., Xu, Q., Liu, H., Phothilimthana, P. M., Wang, S., Goldie, A., Mirhoseini, A., and Laudon, J. (2020). Transferable graph optimizers for ml compilers.

Zhou, Y., Zhang, R., Chen, C., Li, C., Tensmeyer, C., Yu, T., Gu, J., Xu, J., and Sun, T. (2021). LAFITE: towards language-free training for text-to-image generation.

Zhu, M., Pan, P., Chen, W., and Yang, Y. (2019). DM-GAN: dynamic memory generative adversarial networks for text-to-image synthesis.

Zhu, X., Yao, J., and Huang, J. (2016). Deep convolutional neural network for survival analysis with pathological images. In *2016 IEEE International Conference on Bioinformatics and Biomedicine (BIBM)*, pages 544–547. IEEE.

Zhu, Y., Kiros, R., Zemel, R., Salakhutdinov, R., Urtasun, R., Torralba, A., and Fidler, S. (2015). Aligning books and movies: Towards story-like visual explanations by watching movies and reading books. In *Proceedings of the IEEE international conference on computer vision*, pages 19–27.

Zhuang, C., Yan, S., Nayebi, A., Schrimpf, M., Frank, M. C., DiCarlo, J. J., and Yamins, D. L. (2021). Unsupervised neural network models of the ventral visual stream. 118(3):e2014196118.