Multimodal Transformer

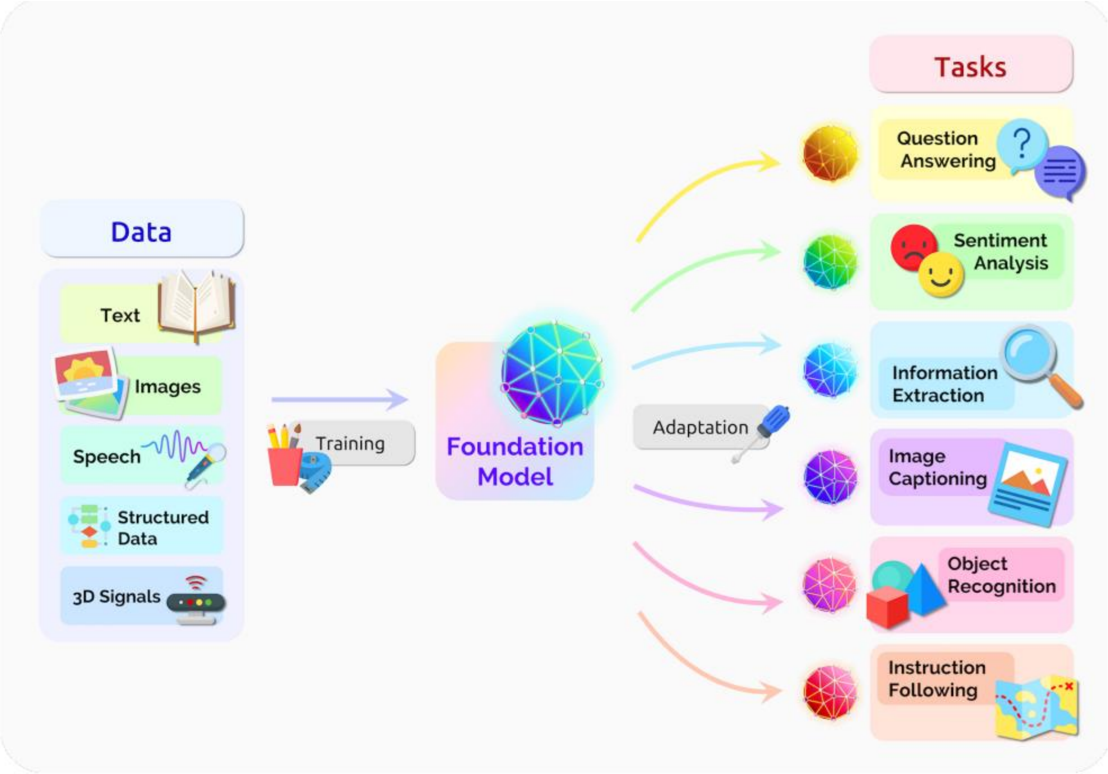
COV884: Special Module In AI
Shivangi Bithel

Topics

- Foundational Model
- Vision-language pre-trained models
  - UNITER
  - VILT
  - FLAVA
- Unified Transformer
- Multimodal Transformer
  - Perceiver
  - Perceiver IO
- Reviews

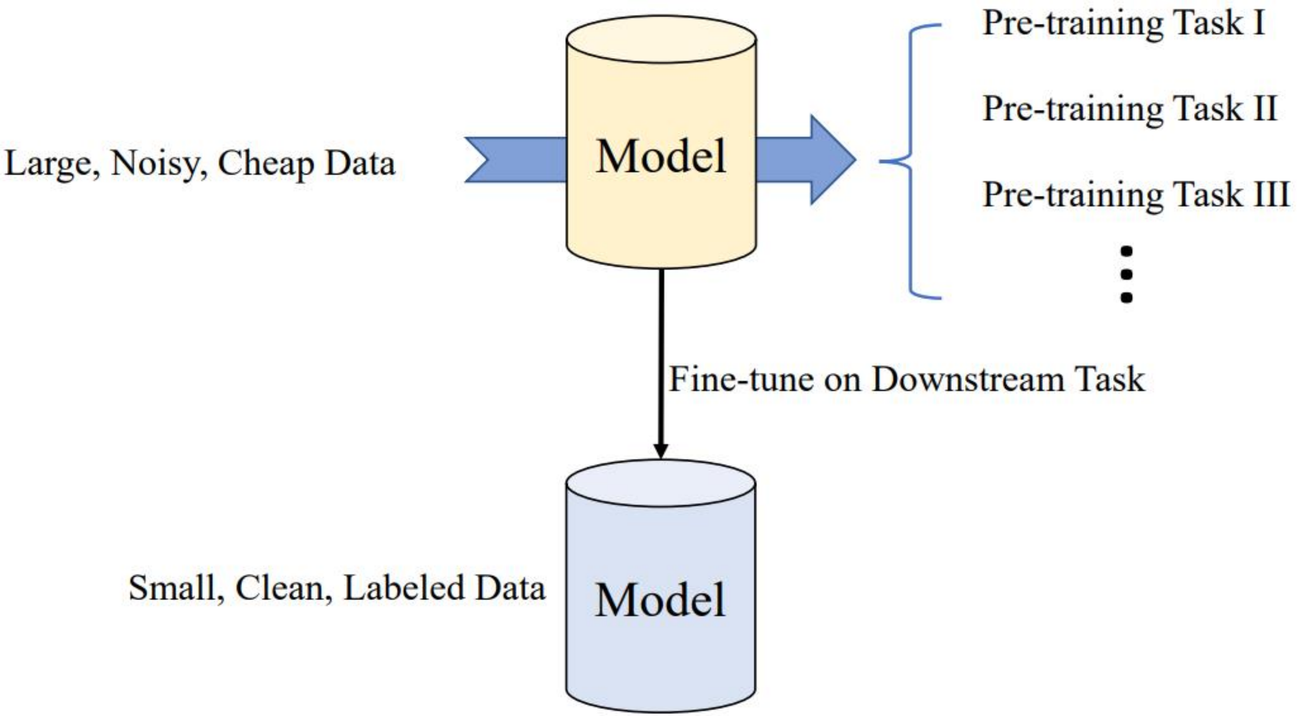
Foundational Models

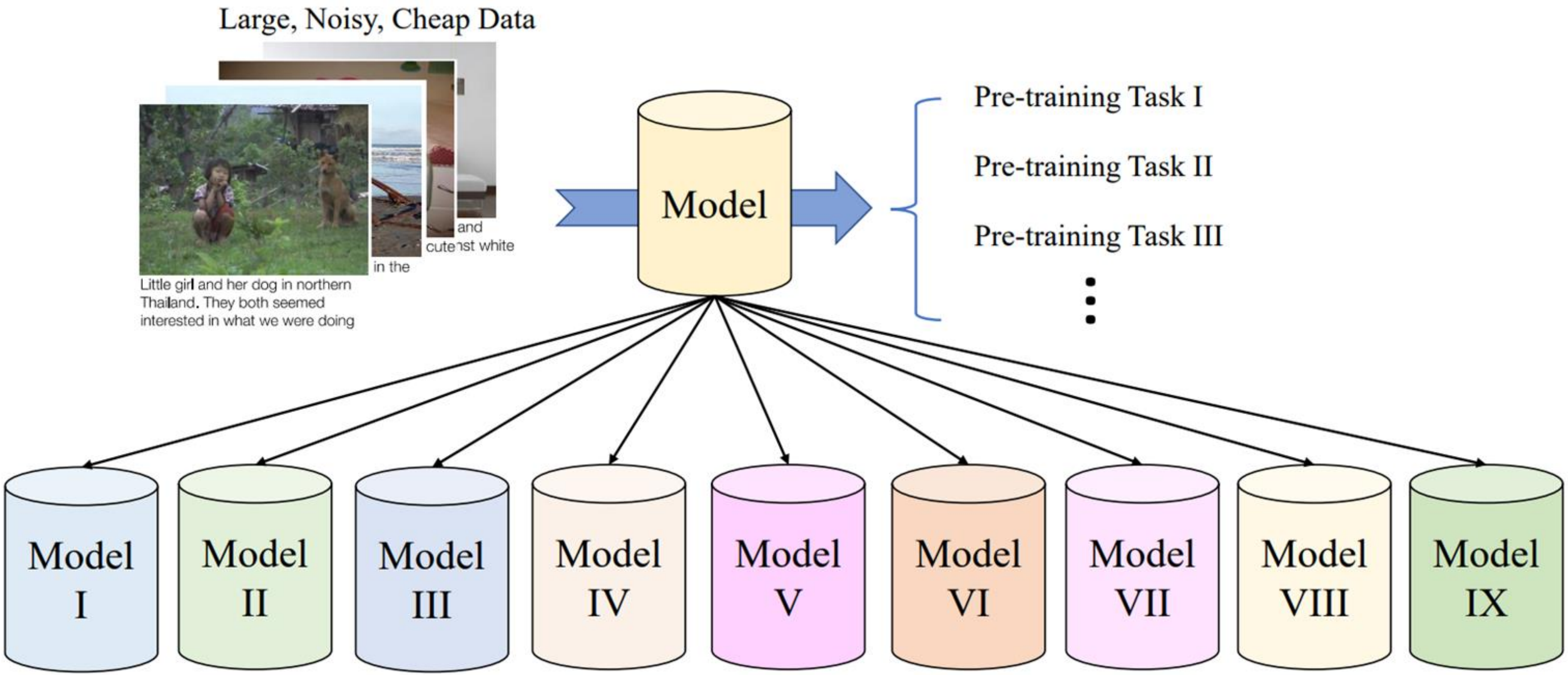
A foundation model can centralize the information from all the data from various modalities. This one model can then be adapted to a wide range of downstream tasks.



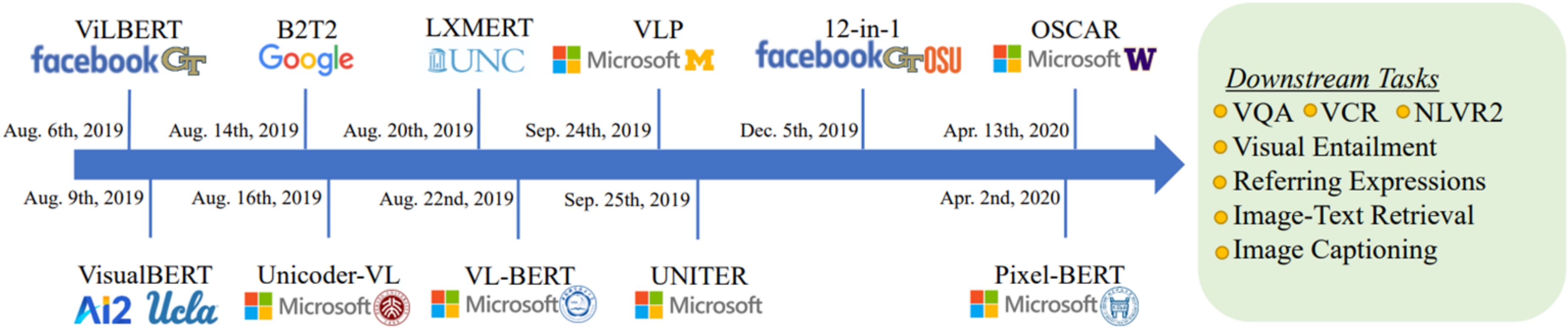
On the Opportunities and Risks of Foundation Models
https://arxiv.org/pdf/2108.07258.pdf

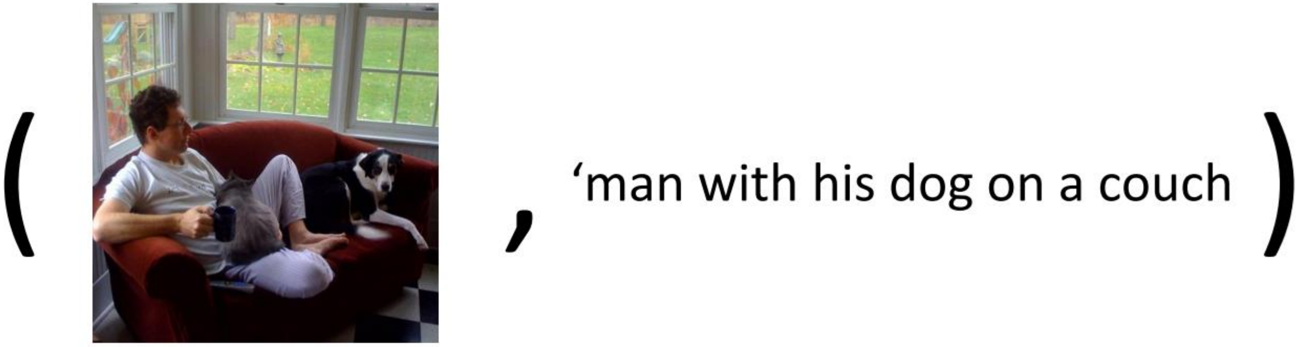
Self-supervised Learning for Vision-and-Language





Large, Noisy, Cheap Data
Little girl and her dog in northern Thailand. They both seemed interested in what we were doing
in the
cute
and
Model
Pre-training Task I
Pre-training Task II
Pre-training Task III
Model I
Model II
Model III
Model IV
Model V
Model VI
Model VII
Model VIII
Model IX


ViLBERT
Aug. 6th, 2019
B2T2
Aug. 14th, 2019
LXMERT
Aug. 20th, 2019
VLP
Sep. 24th, 2019
12-in-1
Dec. 5th, 2019
OSCAR
Apr. 13th, 2020
Aug. 9th, 2019
VisualBERT
Aug. 16th, 2019
Unicoder-VL
Aug. 22nd, 2019
VL-BERT
Sep. 25th, 2019
UNITER
Apr. 2nd, 2020
Pixel-BERT
Downstream Tasks
VQA
VCR
NLVR2
Visual Entailment
Referring Expressions
Image-Text Retrieval
Image Captioning


(, 'man with his dog on a couch)

Common Pre-training datasets

COCO

A close up view of a pizza sitting on a table with a soda in the back.

Visual Genome

a lenovo laptop rebooting

S

Fro
13,
in fi

DATASET	SIZE	Avg Text Length
COCO	0.9M	12.4
SBU Captions	1M	12.1
Localized Narratives	1.9M	13.8
Conceptual Captions	3.1M	10.3
Visual Genome	5.4M	5.1
Wikipedia Image Text	4.8M	12.8
Conceptual Captions 12M	11M	17.3
Red Caps	11.6	9.5
YFCC100M	30.3M	12.7
FLICKR30K	31K	16.6
CLIP	400M	
ALIGN	1.8B	
FLIP	300M	

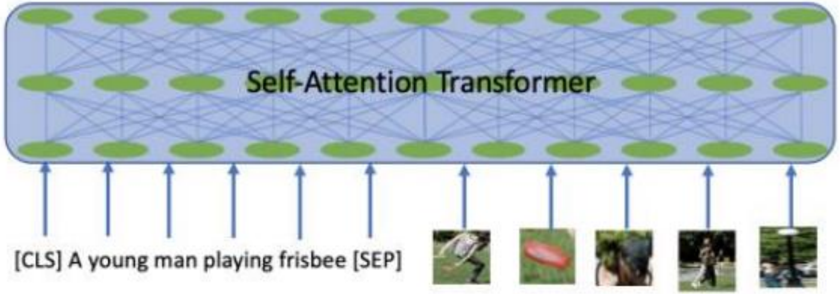
CC12M

Jumping girl in a green summer dress stock illustration

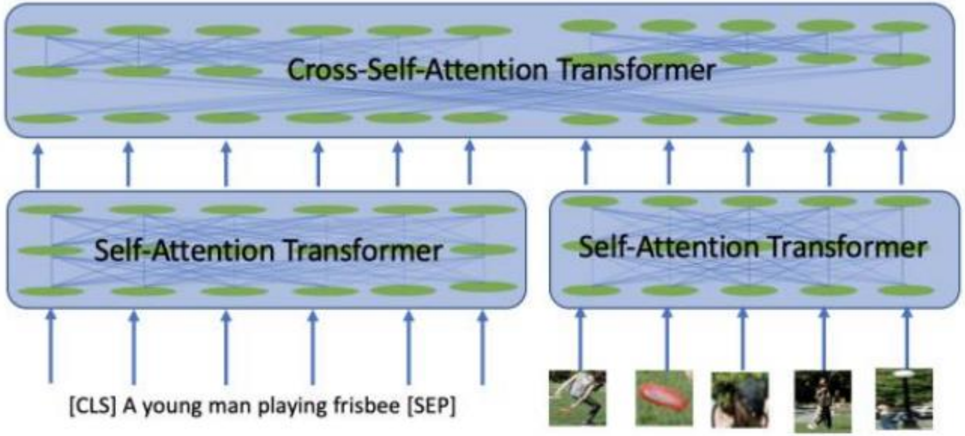
YFCC filtered

In the kitchen at the Muse Nissim de Camondo

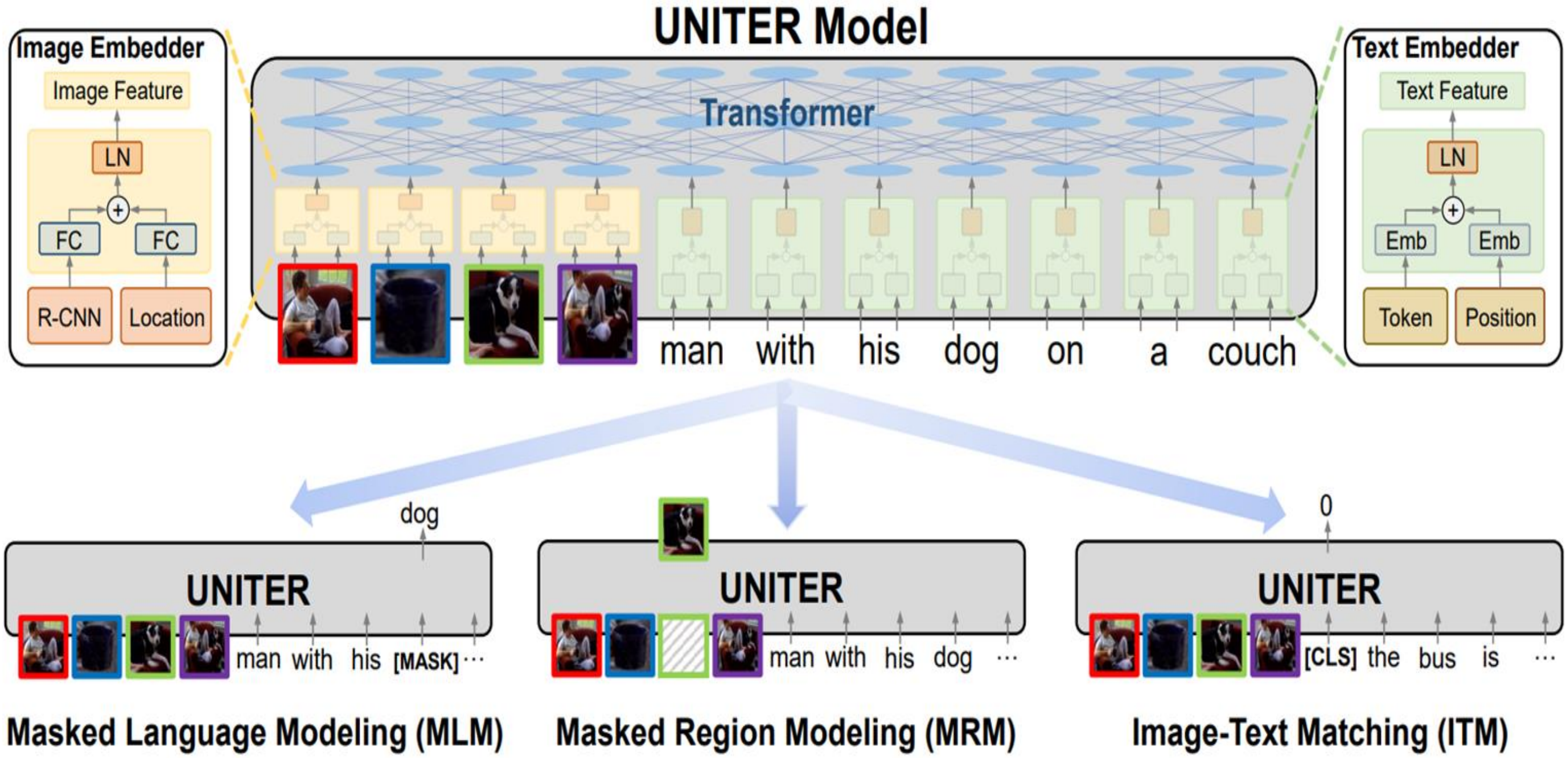
Model Architecture



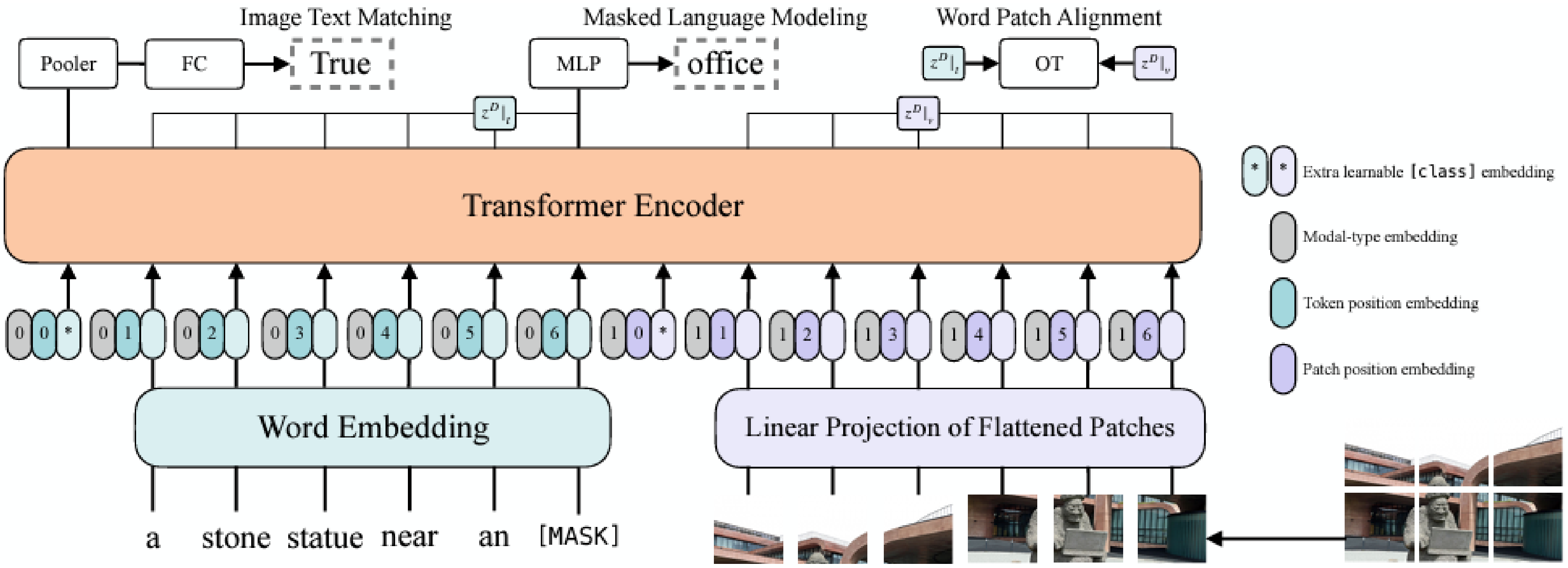
(a) Single-stream Model.



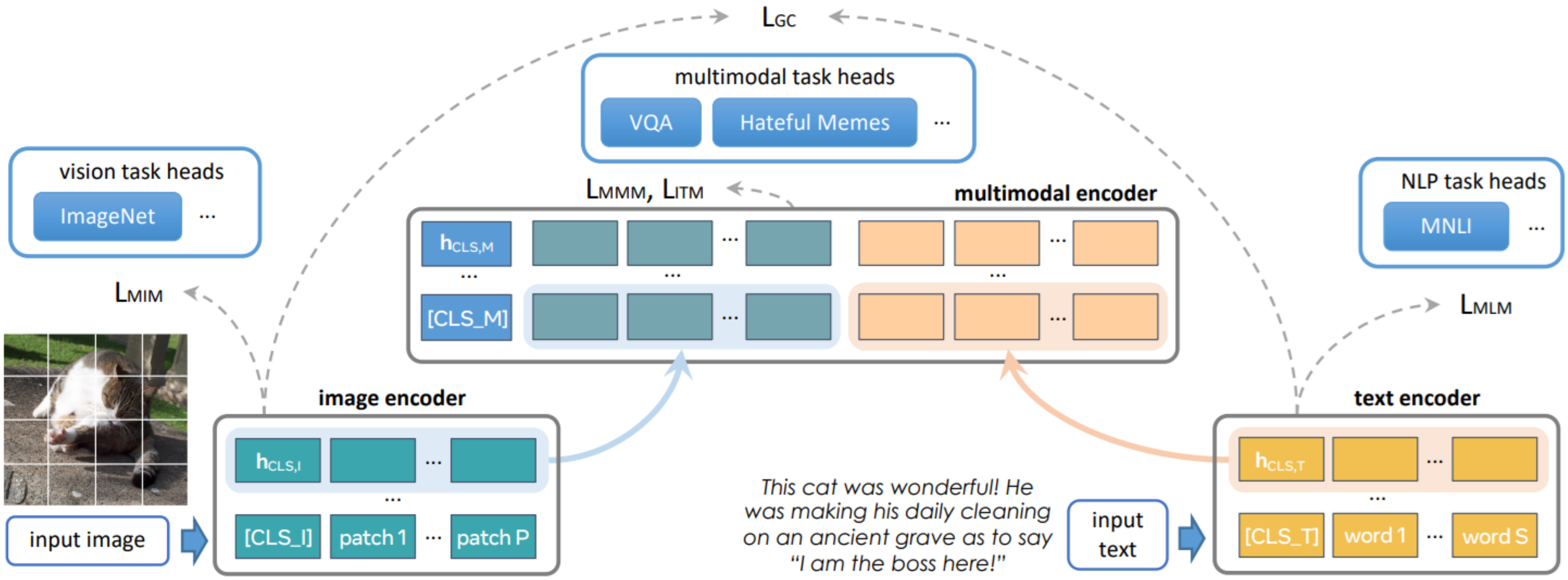
(b) Two-stream Model.


UNITER Model
Image Embedder
Image Feature
LN
FC
FC
R-CNN
Location
Transformer
man with his dog on a couch
Text Embedder
Text Feature
LN
Emb
Emb
Token
Position
dog
UNITER
man with his [MASK] ···
Masked Language Modeling (MLM)
UNITER
man with his dog ···
Masked Region Modeling (MRM)
0
UNITER
[CLS] the bus is ···
Image-Text Matching (ITM)

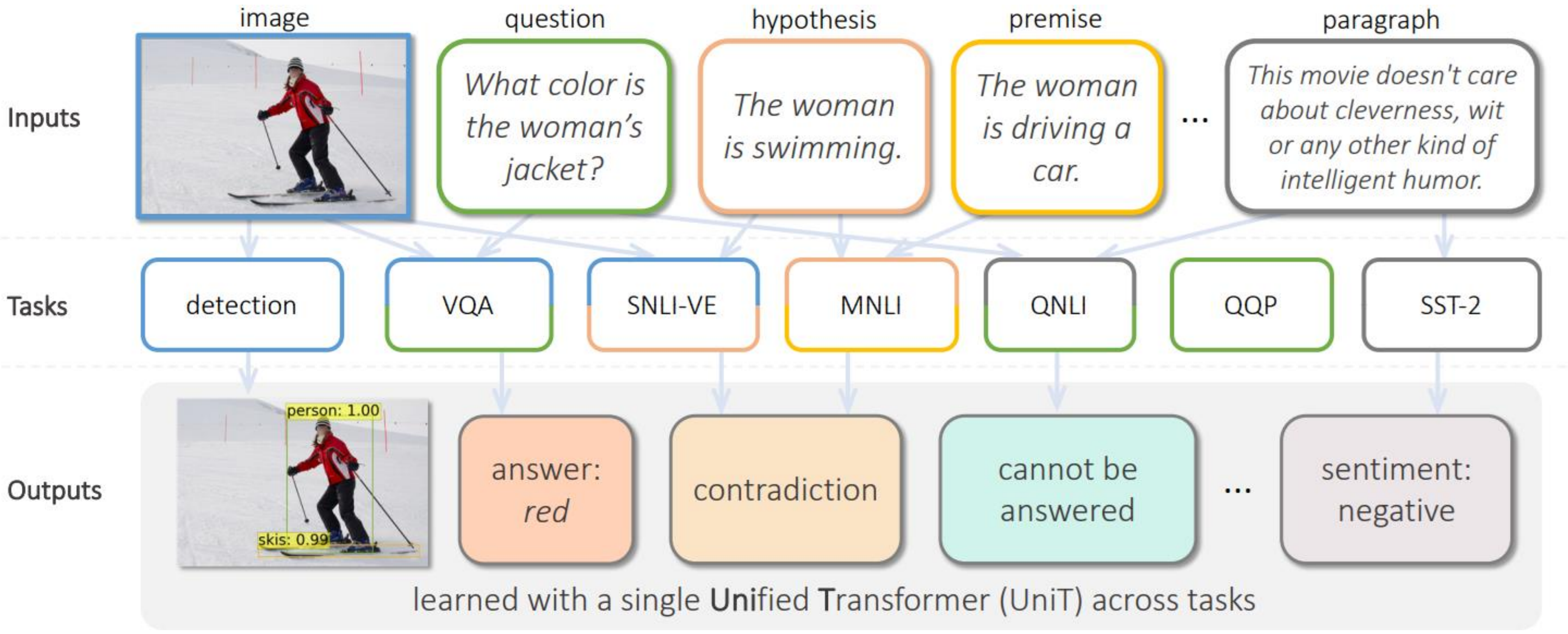
ViLT: Vision-and-Language Transformer Without Convolution or Region Supervision

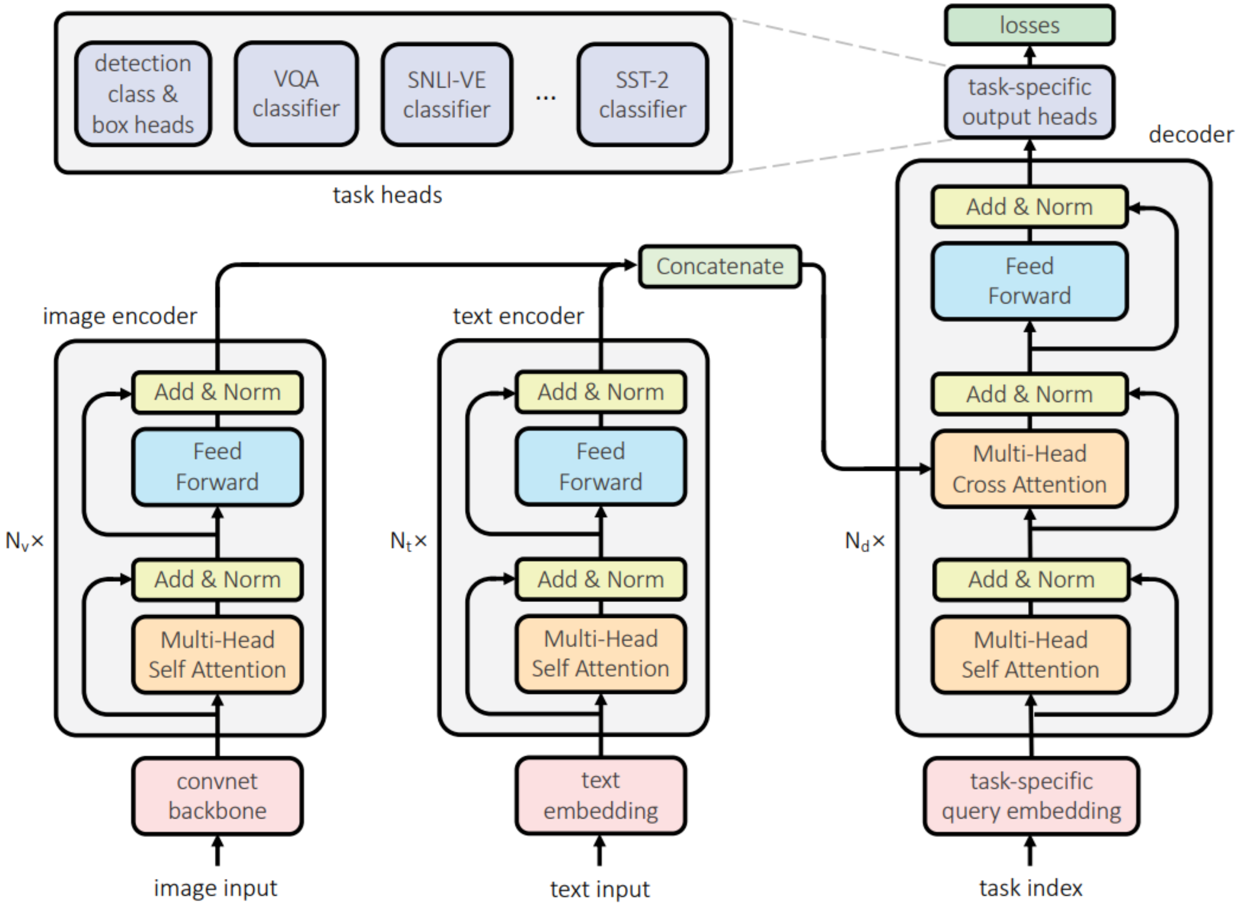


FLAVA : A Foundational Language And Vision Alignment Model



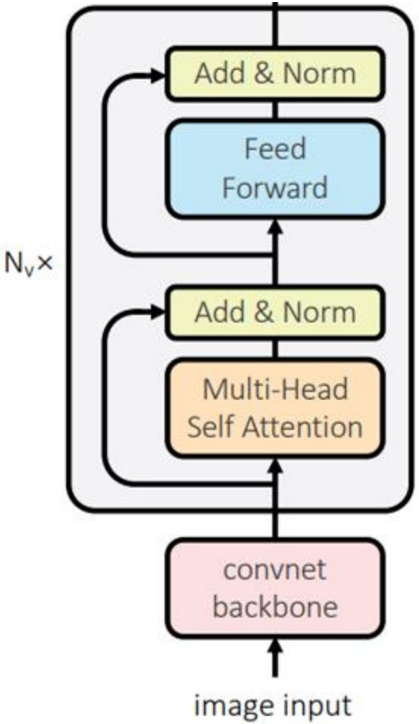
UniT: Multimodal Multitask Learning with a Unified Transformer





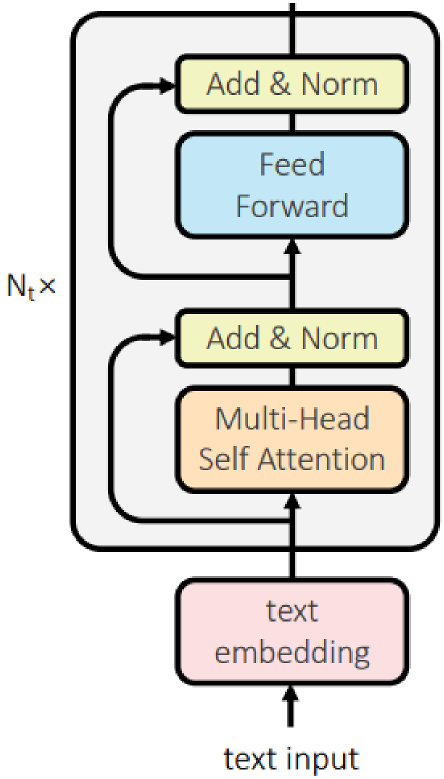
detection class & box heads
VQA classifier
SNLI-VE classifier
...
SST-2 classifier
task heads
losses
task-specific output heads
decoder
Add & Norm
Feed Forward
Add & Norm
Multi-Head Cross Attention
Add & Norm
Multi-Head Self Attention
Concatenate
image encoder
text encoder
Add & Norm
Feed Forward
Add & Norm
Multi-Head Self Attention
$N_v\times$
$N_t\times$
$N_d\times$
convnet backbone
text embedding
task-specific query embedding
image input
text input
task index

Image encoder

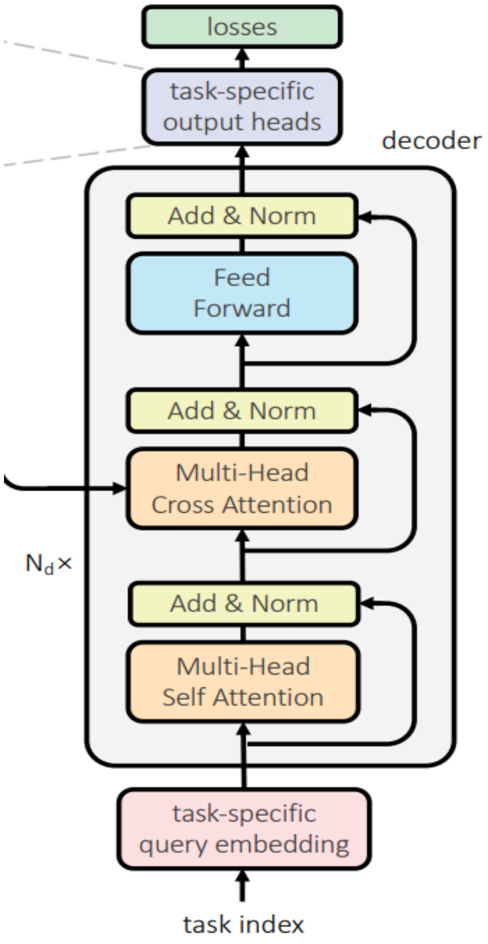


To encode the input image I, the encoder first uses a convolutional neural network followed by a transformer encoder and generates output into a list of encoded visual hidden states $h^v = \{h^v_1, h^v_2, \cdots, h^v_L\}$.

Text encoder



To encode the input text T, the encoder uses a transformer encoder and generates output into a list of encoded textual hidden states $h^t = \{h^t_1, h^t_2, \cdots, h^t_S\}$.



Domain-agnostic UniT decoder and Task-specific output heads

The same decoder is used to perform unimodal and multimodal tasks. In case of Image only tasks the input to the decoder is $h^{enc} = h^{v}$, in case of text only task the input to the decoder is $h^{enc} = h^{t}$, and in case of multimodal tasks the input to the decoder in $h^{enc} = concat(h^{v},h^{t})$.

- The transformer decoder D takes the encoded input sequence h^{enc} and a task-specific query embedding sequence q^{task} of length q. It outputs a sequence of decoded hidden states $h^{dec,l}$ for each of the *l*-th transformer decoder layer, which has the same length q as the query embedding q^{task}.

$$\mathbf{h^{dec,\mathit{l}} = D(h^{enc}\ ,\ q^{task})}$$

- The decoder architecture follows the transformer decoder implementation in DETR. In the *l*-th decoder layer, self-attention is applied among the decoder hidden states $h^{dec,l}$ at different positions and cross-attention is applied to the encoded input modalities h^{enc} .
- A task-specific prediction head is applied over the decoder hidden states $\{h^{dec,l}\}$ for each task *t*.

Training Details

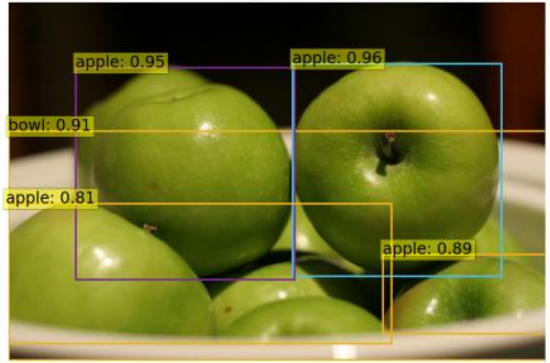
UniT is jointly trained on multiple tasks. At each iteration during training, model randomly selects a task and a dataset to fill a batch of samples. Authors manually specified the sampling probability for each task based on the dataset size and empirical evidence.

Datasets used - MSCOCO, Visual Genome (VG), GLUE benchmark: QNLI, QQP, MNLI-mismatched, and SST-2, VQAv2 dataset and SNLI-VE dataset

Exact training details are mentioned in the paper for reference.

TASKS

object detection (COCO det.)



object detection (VG det.)

visual question answering (VQAv2)

question: How are the zebras related?
answer: mother and child

question: Which food contains the most potassium?
answer: banana

visual entailment (SNLI-VE)

hypothesis: A man with a sweatshirt is in a wooded area.
prediction: entailment

hypothesis: Two dogs are sleeping in the grass.
prediction: contradiction

GLUE TASKS

QNLI

paragraph: As of that day, the new constitution heralding the Second Republic came into force.
question: What came into force after the new constitution was herald?
prediction: answerable

paragraph: For example, Joseph Haas was arrested for allegedly sending an email to the Lebanon, New Hampshire city councilors stating, "Wise up or die."
question: What year did the the case go before the supreme court?
prediction: cannot be answered

MNLI-mm

premise: Captain Victor Saracini and First Officer Michael Horrocks piloted the Boeing 767, which had seven flight attendants.
hypothesis: The Captain was Michael Horrocks and there were 4 flight attendants aboard.
prediction: contradiction

premise: They were promptly executed.
hypothesis: They were executed immediately upon capture.
prediction: neutral

QQP

question 1: Is there a reason why we should travel alone?
question 2: What are some reasons to travel alone?
prediction: equivalent

question 1: Why was the Roman Empire so successful?
question 2: What are some of the rarely known facts about the Roman Empire?
prediction: not equivalent

SST-2

paragraph: allows us to hope that nolan is poised to embark a major career as a commercial yet inventive filmmaker.
sentiment: positive

paragraph: in its best moments , resembles a bad high school production of grease , without benefit of song.
sentiment: negative

Multitask learning on detection and VQA

decoder setup	COCO det. mAP	VG det. mAP	VQAv2 accuracy
single-task training	40.6 / –	3.87	66.38 / –
separate	**40.8** / –	3.91	**68.84** / –
shared	37.2 / –	4.05	68.79 / –
shared (COCO init.)	**40.8** / 41.1	**4.53**	67.30 / 67.47

training data	COCO det. mAP	VG det. mAP	VQAv2 accuracy
single-task training	40.6	3.87	66.38
COCO + VQAv2	40.2	–	66.88
VG + VQAv2	–	3.83	**68.49**
COCO + VG + VQAv2	**40.8**	**4.53**	67.30

Three settings of decoder here are

1. separate decoders on different tasks
2. single shared decoder for all tasks
3. Coco detection initialized before training on joint tasks

In this experiment, only one dataset is being used from each task i.e. either COCO or Visual Genome from Object detection task is used.

Unified Transformer for multiple domains

#	decoder setup	COCO det. mAP	VG det. mAP	VQAv2 accuracy	SNLI-VE accuracy	QNLI accuracy	MNLI-mm accuracy	QQP accuracy	SST-2 accuracy
1	UniT – single-task training	40.6	3.87	66.38 / –	70.52 / –	91.62 / –	84.23 / –	91.18 / –	91.63 / –
2	UniT – separate	32.2	2.54	67.38 / –	74.31 / –	87.68 / –	81.76 / –	90.44 / –	89.40 / –
3	UniT – shared	33.8	2.69	67.36 / –	74.14 / –	87.99 / –	81.40 / –	90.62 / –	89.40 / –
4	UniT – separate (COCO init.)	38.9	3.22	67.58 / –	74.20 / –	87.99 / –	81.33 / –	90.61 / –	89.17 / –
5	UniT – shared (COCO init.)	39.0	3.29	66.97 / 67.03	73.16 / 73.16	87.95 / 88.0	80.91 / 79.8	90.64 / 88.4	89.29 / 91.5
6	UniT – per-task finetuning	42.3	4.68	67.60 / –	72.56 / –	86.92 / –	81.53 / –	90.57 / –	88.06 / –
7	DETR [5]	43.3	4.02	–	–	–	–	–	–
8	VisualBERT [31]	–	–	67.36 / 67.37	75.69 / 75.09	–	–	–	–
9	BERT [14] (bert-base-uncased)	–	–	–	–	91.25 / 90.4	83.90 / 83.4	90.54 / 88.9	92.43 / 93.7

The experiment is on three different settings:
(i) single-task training where each model is trained separately on each task,
(ii) multi-task training with separate decoders where the model has a specific decoder for each task but is jointly trained on all of the tasks, and
(iii) multi-task training same as (ii) but with a shared decoder instead of separate ones.

Ablation analyses with different configurations on COCO detection, SNLI-VE, and MNLI.

#	Model configuration	COCO det. mAP	SNLI-VE accuracy	MNLI-mm accuracy
1	UniT (default, d_t^d=768, N_d=6)	38.79	69.27	81.41
2	decoder layer number, N_d=8	40.13	68.17	80.58
3	decoder layer number, N_d=12	39.02	68.82	81.15
4	decoder hidden size, d_t^d=256	36.32	69.68	81.09
5	using all hidden states from BERT instead of just [CLS]	38.24	69.76	81.31
6	losses on all decoder layers for SNLI-VE and MNLI-mm	39.46	69.06	81.67
7	no task embedding tokens	38.61	70.22	81.45
8	batch size = 32	35.03	68.57	79.62

Paper of the day:

- Perceiver: General Perception with Iterative Attention
- PERCEIVER IO: A General Architecture for Structured Inputs & Outputs

Perceiver: General Perception with Iterative Attention

Andrew Jaegle [1] Felix Gimeno [1] Andrew Brock [1] Andrew Zisserman [1] Oriol Vinyals [1] Joao Carreira [1]

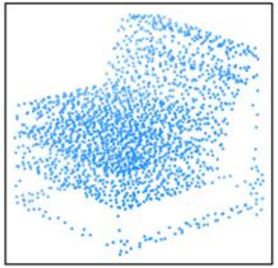
Input data

Image

ImageNet

Video = Image + Audio

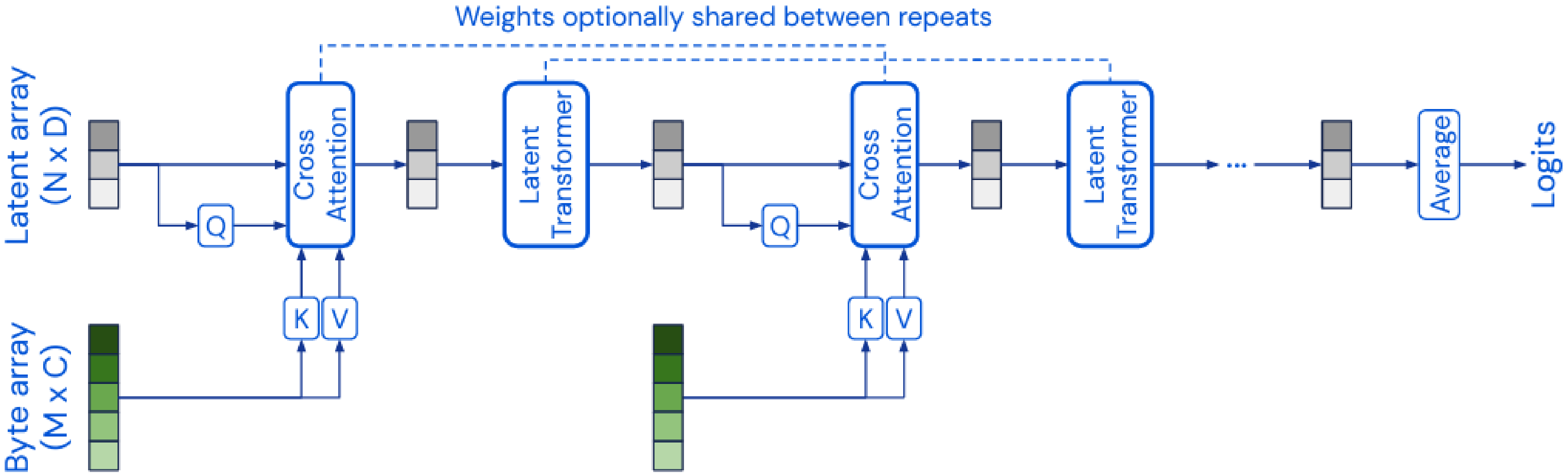
AudioSet



3D Point clouds

ModelNet40

The Perceiver Architecture



Positional Encoding - more domain specific or generic?

1. Following the idea that greater generality follows from making as much of a system learnable as possible - we are using feature based approach rather than hardcoding the values of positions.
2. Designing an efficient way of providing these positional encoding is time consuming (as we have seen in TAPAS paper the encoding for tabular data) - but using fourier features, which can adapt to new domain and modality easily makes the work easy.
3. In case of multimodal data like video, where image and audio is given simultaneously, the learned positional encodings can also learn to distinguish between these different modalities.

Tasks and Results

1. Image Classification on ImageNet
    - ImageNet is a unilabel dataset - every image belongs to a single class
    - Loss function used to train the classification task - Cross Entropy
    - Output - softmax over the logits
    - Optimizer - LAMB
    - Top-1 accuracy

	Raw	Perm.	Input RF
ResNet-50 (FF)	73.5	39.4	49
ViT-B-16 (FF)	76.7	61.7	256
Transformer (64x64) (FF)	57.0	57.0	4,096
Perceiver:			
(FF)	78.0	78.0	50,176
(Learned pos.)	70.9	70.9	50,176

2. Audio and Video → AudioSet

- Audio Event Classification in Video - Videos can have multiple labels
- Loss function - Sigmoid Cross entropy loss
- Evaluation: Mean Average Precision → mAP
- Near SOTA results

Model / Inputs	Audio	Video	A+V
Benchmark (Gemmeke et al., 2017)	31.4	-	-
Attention (Kong et al., 2018)	32.7	-	-
Multi-level Attention (Yu et al., 2018)	36.0	-	-
ResNet-50 (Ford et al., 2019)	38.0	-	-
CNN-14 (Kong et al., 2020)	43.1	-	-
CNN-14 (no balancing & no mixup) (Kong et al., 2020)	37.5	-	-
G-blend (Wang et al., 2020c)	32.4	18.8	41.8
Attention AV-fusion (Fayek & Kumar, 2020)	38.4	25.7	46.2
Perceiver (raw audio)	38.3	25.8	43.5
Perceiver (mel spectrogram)	38.4	25.8	43.2
Perceiver (mel spectrogram - tuned)	-	-	44.2

3. 3D Point cloud - Object Classification task

- Convert 3D point cloud → 2D Grid and then feed it through the model
- SOTA here is carefully designed model with sophisticated data augmentation and feature engineering Procedure. Perceiver still Beats the generic ImageNet baselines

	Accuracy
PointNet++ (Qi et al., 2017)	**91.9**
ResNet-50 (FF)	66.3
ViT-B-2 (FF)	78.9
ViT-B-4 (FF)	73.4
ViT-B-8 (FF)	65.3
ViT-B-16 (FF)	59.6
Transformer (44x44)	82.1
Perceiver	**85.7**

Problems yet to be solved

- The Model doesn't always do as well as models made for a particular modality.
- There is a possibility of overfitting in the perceiver model as the dataset is not large enough while the model is quite big to memorize the data points. This creates a scope for trying pre-trained models with large amounts of data.
- The model still employs the modality-specific augmentation and position encoding
- At this point, Perceiver doesn’t exhibit any kind of cross-modal tasks.

PERCEIVER IO: A GENERAL ARCHITECTURE FOR STRUCTURED INPUTS & OUTPUTS

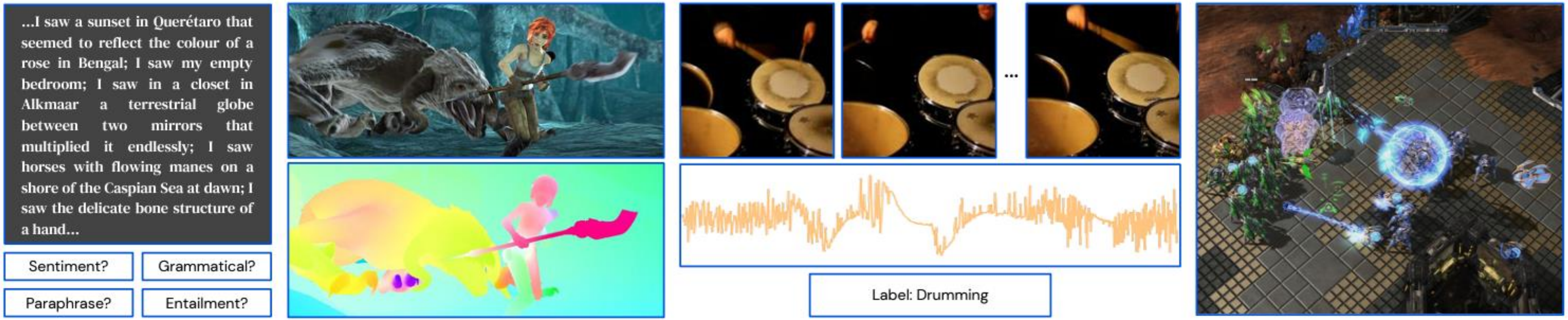
Andrew Jaegle, Sebastian Borgeaud, Jean-Baptiste Alayrac, Carl Doersch, Catalin Ionescu,

David Ding, Skanda Koppula, Daniel Zoran, Andrew Brock, Evan Shelhamer, Olivier Hénaff,

Matthew M. Botvinick, Andrew Zisserman, Oriol Vinyals, João Carreira

DeepMind

Input Data



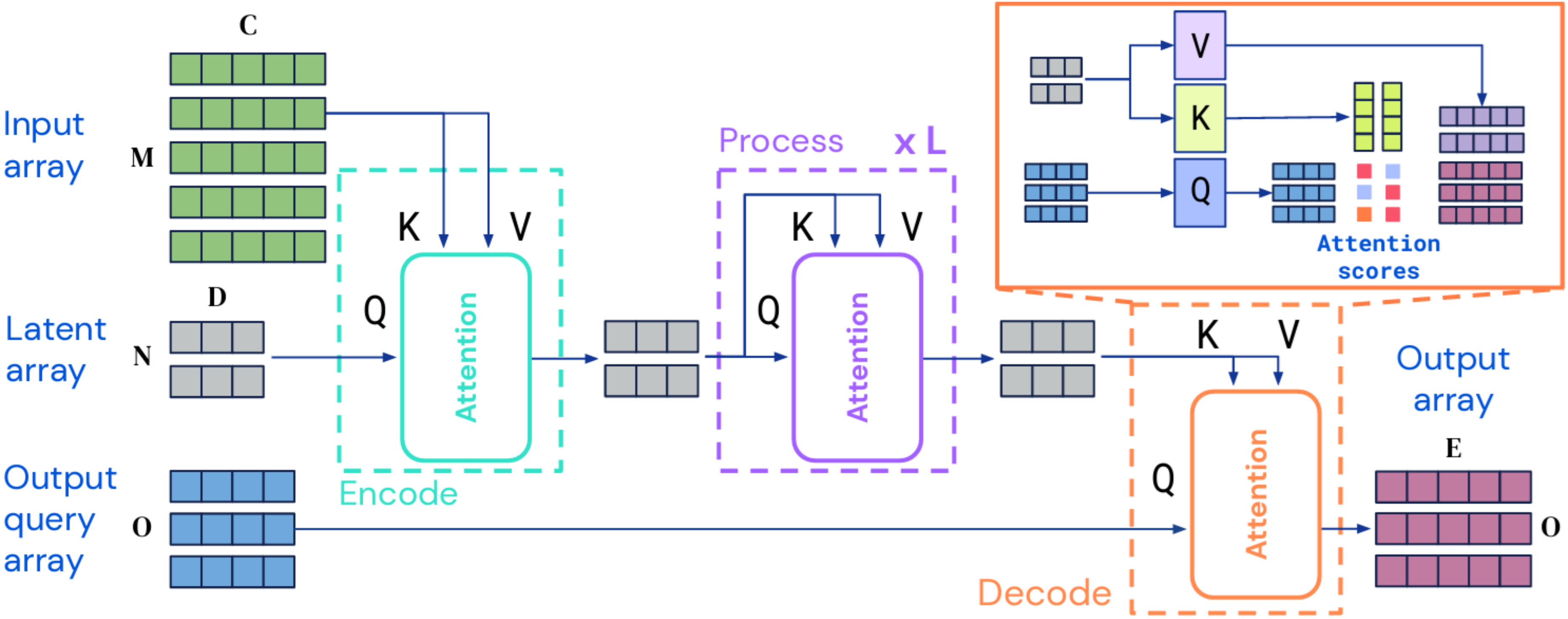
Text → Language Understanding

Image → Optical flow

Video + Audio + class → Multimodal autoencoding

Image → StarCraft II

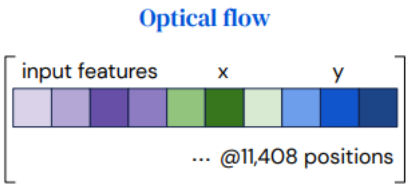
The Perceiver IO Architecture

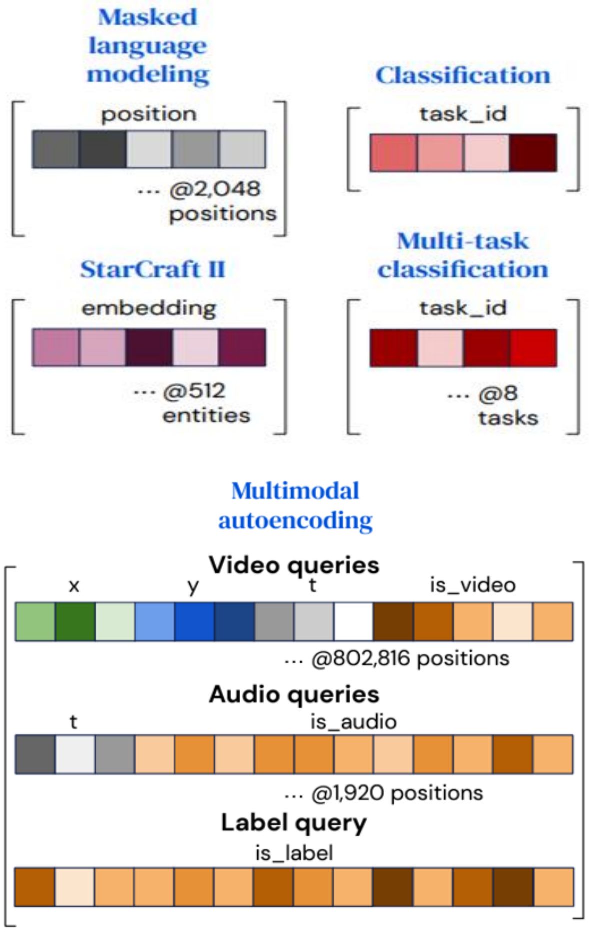


Query Construction

The queries are constructed with output-specific features to produce outputs with different semantics.

- **Language** - each output point differs only in its position → a position embedding can be used.
- **StarCraft II** - Input features for the target output alone
- **Optical flow** - Input features for the target output along with position embeddings
- **Multi-{task, modal}** - use one embedding for each {task, modality} instead of each position.
- **Classification tasks** - embedding can be learned and reused
- **Multimodal autoencoding** - features that are specific to some queries (like xy position) can be combined with
ch also pad embeddings to fixed





Experiments - LANGUAGE

Model	Tokenization	M	N	Depth	Params	FLOPs	SPS	Avg.
BERT Base (test)	SentencePiece	512	512	12	110M	109B	-	81.0
BERT Base (ours)	SentencePiece	512	512	12	110M	109B	7.3	81.1
Perceiver IO Base	SentencePiece	512	256	26	223M	119B	7.4	**81.2**
BERT (matching FLOPs)	UTF-8 bytes	2048	2048	6	20M	130B	2.9	71.5
Perceiver IO	UTF-8 bytes	2048	256	26	201M	113B	7.6	81.0
Perceiver IO++	UTF-8 bytes	2048	256	40	425M	241B	4.2	**81.8**

The avg(average) denotes the average performance on the glue benchmark datasets and tasks. We can observe that with comparable FLOPs, the depth and number of parameters that perceiverIO can use increases which further increases the understanding of the model and thus better results in comparison to dedicated architecture of BERT.

Architectural Details

Model	BERT Base	BERT matching FLOPs	Perceiver IO Base	Perceiver IO	Perceiver IO++
Tokenizer	SentencePiece	UTF-8 bytes	SentencePiece	UTF-8 bytes	UTF-8 bytes
Number of inputs (M)	512	2048	512	2048	2048
Input embedding size (C)	768	768	768	768	768
Number of Process layers	12	6	26	26	40
Number of latents (N)	-	-	256	256	256
Latent size (D)	-	-	1280	1280	1536
FFW hidden dimension for latents	-	-	1280	1280	1536
Number of output queries during pretraining (O)	-	-	512	2048	2048
Dimension of learned queries (E)	-	-	768	768	768
FFW hidden dimension for outputs	-	-	768	768	768
Training steps/second	7.3	2.9	7.4	7.6	4.2

These are the hyperparameter details for the language understanding experiment.

Full GLUE results

Model	Tokenizer	Multi-task	CoLA	MNLI-m/mm	MRPC	QNLI	QQP	RTE	SST-2	STS-B	Average
Bert Base (test) (Devlin et al., 2019)	SentencePiece	No	52.10	84.60/83.40	84.80	90.50	89.20	66.40	93.50	87.10	80.95
Bert Base (ours)	SentencePiece	No	50.28	85.56/85.68	85.75	92.67	91.05	61.72	93.98	88.04	81.14
Perceiver IO Base	SentencePiece	No	47.11	84.53/85.03	87.25	92.12	90.22	65.23	94.38	88.18	81.16
BERT (matching FLOPs)	UTF-8 Bytes	No	20.06	74.11/75.55	77.00	85.75	88.23	53.91	89.00	82.84	71.45
Perceiver IO	UTF-8 Bytes	No	50.19	83.22/83.89	87.24	91.71	90.12	64.84	93.17	86.81	80.95
Perceiver IO++	UTF-8 Bytes	No	52.54	84.13/84.91	86.03	92.06	90.46	66.54	93.98	87.93	81.76
Perceiver IO (Shared input token)	UTF-8 Bytes	Yes	47.43	82.03/82.65	89.58	90.18	89.20	82.03	93.17	77.95	81.49
Perceiver IO (Task specific input token)	UTF-8 Bytes	Yes	49.06	82.14/82.64	89.84	90.53	89.40	79.69	93.17	80.02	81.76
Perceiver IO (Multitask query)	UTF-8 Bytes	Yes	47.88	82.05/82.77	90.36	90.37	89.49	80.08	93.75	79.95	81.79

Method	Avg.
Single-task query	81.0
Multitask	
Shared input token	81.5
Task-specific input tokens	**81.8**
Multitask query	**81.8**

There are 4 different settings here:

1. Single task query where the model is trained independently on each task
2. Sharing a single [cls] token among tasks (Shared input token)
3. Using task-specific tokens (Task-specific input token)
4. Use multitask query to finetune on all 8 GLUE tasks simultaneously using the UTF8 byte model

We observe that the multitask approach(4) outperforms single-task approaches and matches the approach that uses 8 task-specific input tokens.

OPTICAL FLOW

Network	Sintel.clean	Sintel.final	KITTI
PWCNet (Sun et al., 2018)	2.17	2.91	5.76
RAFT (Teed & Deng, 2020)	1.95	2.57	**4.23**
Perceiver IO	**1.81**	**2.42**	4.98

Problem Statement - Given two images of the same scene (e.g. two consecutive frames of a video), the task is to estimate the 2D displacement for each pixel in the first image.

Optical flow is challenging for neural networks for two reasons:

- Optical flow relies on finding correspondence: a single frame provides no information about flow, and images with extremely different appearance can produce the same flow.
- Flow is extremely difficult to annotate, and the few datasets with realistic images and high-quality ground truth are small and biased. While it is straightforward to generate large synthetic datasets as training data, e.g. AutoFlow, there is still a large domain gap.

MULTIMODAL AUTOENCODING

Compression Ratio	Audio PSNR	Video PSNR	Top-1 Accuracy
88x	26.97	24.37	10.2%
176x	25.33	24.27	8.6%
352x	14.15	23.21	11.5%

Perceiver IO is evaluated for audio-video-label multimodal autoencoding on the Kinetics700-2020 dataset. The goal of multimodal autoencoding is to learn a model that can accurately reconstruct multimodal inputs in the the presence of a bottleneck induced by an architecture. Perceiver IO pads the inputs with modality-specific embeddings, serialize them into a single 2D input array and query outputs using queries containing position encodings (for video and audio) and modality embeddings.

Table shows the results of Multimodal autoencoding. Higher is better for accuracy and PSNR. These results suggests that Perceiver IO can jointly represent modalities with very different properties.

IMAGE CLASSIFICATION ON IMAGENET

Model	Pretrained?	Accuracy	FLOPs	Params
ConvNet baselines				
ResNet-50 (He et al., 2016)	N	78.6	4.1B	26M
NFNet-F6+SAM (Brock et al., 2021)	N	86.5	377.3B	438.4M
Meta Pseudo Labels (Pham et al., 2021)	Y	90.2	-	480M
ViT baselines				
ViT-B/16 (Dosovitskiy et al., 2021)	N	77.9	55.4B	86M
ViT-H/14 (Dosovitskiy et al., 2021)	Y	88.6	-	632M
DeiT 1000 epochs (Touvron et al., 2021a)	N	85.2	-	87M
CaiT-M48 448 (Touvron et al., 2021b)	N	86.5	329.6B	356M
w/ 2D Fourier features				
Perceiver	N	78.6	404B	42.1M
Perceiver IO, config A	N	79.0	407B	48.4M
Perceiver IO, config B (pretrained)	Y	84.5	213B	212M
w/ learned position features				
Perceiver (learned pos)	N	67.6	404B	55.9M
Perceiver IO, config A (learned pos)	N	72.7	407B	62.3M
w/ 2D conv + maxpool preprocessing				
Perceiver (conv)	N	77.4	367B	42.1M
Perceiver IO, config A (conv)	N	82.1	369B	48.6M
Perceiver IO, config B (conv) (pretrained)	Y	86.4	176B	212M

STARCRAFT II

Entity encoder	Win rate	Params (M)	FLOPs	Train steps/sec
Transformer (Vinyals et al., 2019)	0.87	144	3.3B	2.9
Perceiver IO	0.87	140	0.93B	2.9

To answer the question: “Can Perceiver IO serve as a replacement for a well-tuned Transformer as a symbolic processing engine?” this experiment is performed where Perceiver IO is evaluated on StarCraft II by using it to replace the well-tuned Transformer entity encoder. Perceiver IO matches the performance of the original Transformer despite using fewer FLOPs and parameters and requiring essentially no tuning. Thus we can say that the answer is “YES”.

AUDIOSET

Model	Input	mAP	Latent channels (D)	Params (M)	FLOPs	Train steps/sec
Perceiver	Raw audio + video	42.4	512	21.0	52.3B	3.8
Perceiver IO	Raw audio + video	43.3	512	25.0	52.9B	3.8
Perceiver	mel-spectrogram + video	43.6	512	21.0	60.7B	3.8
Perceiver IO	mel-spectrogram + video	44.9	1024	88.2	129.5B	3.8

Here similar to image classification, we can observe that in case of audio event classification also Perceiver IO with an attention based decoder improves with a small amount in both the settings in comparison to Perceiver.

Conclusion

- From the first paper UniT, we can show that the transformer framework can be applied over a variety of domains to jointly handle multiple tasks within a single unified encoder-decoder model. With a domain-agnostic transformer architecture, the model makes a step towards building general-purpose intelligence agents capable of handling a wide range of applications in different domains, including visual perception, natural language understanding, and reasoning over multiple modalities.
- Owing to the fact that we don’t have time to segregate the data coming from different modalities, we constructed a generic transformer based encoder which can take input in any modality and also produce any structured output with carefully designed queries.

Reviews - Pros

Common:

- Perceiver IO is a a **general architecture capable of handling general-purpose inputs and outputs** across different tasks and modalities. This is very promising to simplify the construction of highly tuned task-specific neural pipelines and improve the multimodal and multi-task problems.
- The proposed architecture is **tested on massive experiments** including language understanding tasks, optical flow, video audio class autoencoding, image classification, and starcraft II and achieves superior performance. Each task is supported with a detailed ablation study to shed light on future research.
- **FLOPs is used as a metric** - contrasting views

Cons

- In table 1, the Perceiver IO Base has 119B FLOPs and the BERT model they are comparing with has 109B FLOPs. I am not really sure if a **difference of 10B FLOPs** fall in the comparable range. Also the former is more than twice the size of the BERT Base model (wrt parameters), so that might be the case of better performance (though the idea that FLOPs matter more than parameters is intuitive) (JAI)
- Though they show using byte format performs better, I believe the insight tokenisation provides in the domain of language is valuable and cannot be captured by bytes. The **bytes model cannot be directly compared with the tokenized model due to mismatch of number of parameters.** (Shreya)

- For language-based experiments, **Model has been compared with BERT, but the pre-training data is different**. In particular, Perceiver IO also uses the C4 dataset used in the T5 paper which is quite clean. This seems to be unfair for BERT while comparing the 2 models. Also, comparisons should be made with other models like Roberta, T5 (especially because they are using the same data as T5), so that the reader comes to know the complete picture. (This is also a disagreement with Jai, that they don't have a performance hit when they don't use tokenizers. Maybe, they have a performance hit, but are improving it by using more data, or other engineering tricks? (Harman)
- Training on relatively simple domains (like imagenet) becomes very expensive with Perceiver. **FLOPs required are very large (~10x) compared to Vision transformers**. (Harman)

- Model **understanding(explainability) would be very difficult** in such settings. (Rohit)
- PerceiverIO is general enough as a computation backbone. But it does not fully disentangles task specific modeling. Previous models use encoders and/or decoders targeted towards capturing specific structure in the data. This effort has now been pushed towards designing of the inputs. Though it seems much easier in PerceiverIO for example simple byte vocabulary worked for MLM. (Vishal)
- Why do authors say that **FLOPs matter more** that number of parameter? This may be true during training but need not be true during inference. More parameters means more memory. (Vishal)
- There hasn't been any study of performance with size-change in an **intra-task setting**. If the architecture could handle changes in image dimensions, say, that would be very interesting.

Extension and Future work

- Multitasking with multiple domains as done by UniT
- Adaptation to Multimodal and crossmodal tasks like image caption, cross-modal retrieval, VQA, etc.
- Here model hyper-parameters are task specific - shared parameters across all tasks can be one direction to work
- Using Graph as input modality
- Multilingual data within language data like Chinese
- Explainability of the model
- Adaptation to Zero-shot settings
- Can release the model in different sizes

THANK YOU!!