Perceiver: General Perception with Iterative Attention

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

이관호

Motivation

- Perceptron model used in deep learning designed for individual modalities, often relying on domain specific assumptions (e.g. local grid structures exploited by all existing vision models)

- These priors introduce helpful inductive biases, but also lock models to individual modalities.

  But, given the increasing availability of large datasets, is the choice to bake such biases into our models with hard architectural decision the correct one?

Motivation (2)

- Transformers are very flexible architectural blocks that make few assumptions about their inputs.

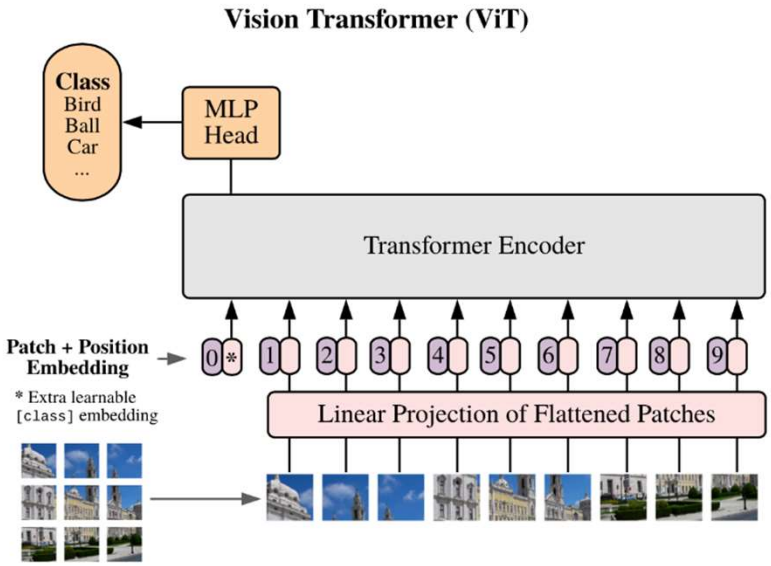
- But Transformers scale quadratically with the number of inputs, in terms of both memory and computation

Perceiver introduce a small set of latent units that forms an attention bottleneck.

> This eliminates the quadratic scaling problem. (quadratic to linear in input size)

Related work

- Vision Transformer (ViT, Alexey Dosovitskiy et al.)
  1) Divide images into patches
  2) put patches into CNN (Resnet) and get feature map
  3) flatten and put into transformer encoder



But Vit relies on pixels' grid structure to reduce computational complexity

Related work

- Set Transformer(Juho Lee(Kaist) et al.)

$$\mathrm{MAB}(X, Y) = \mathrm{LayerNorm}(H + \mathrm{rFF}(H)), \quad (6)$$
$$\text{where } H = \mathrm{LayerNorm}(X + \mathrm{Multihead}(X, Y, Y; \omega)), \quad (7)$$

Induced Set Attention Block (ISAB) which bypasses scaling problem.

$$\mathrm{ISAB}_m(X) = \mathrm{MAB}(X, H) \in \mathbb{R}^{n \times d}, \quad (9)$$
$$\text{where } H = \mathrm{MAB}(I, X) \in \mathbb{R}^{m \times d}. \quad (10)$$

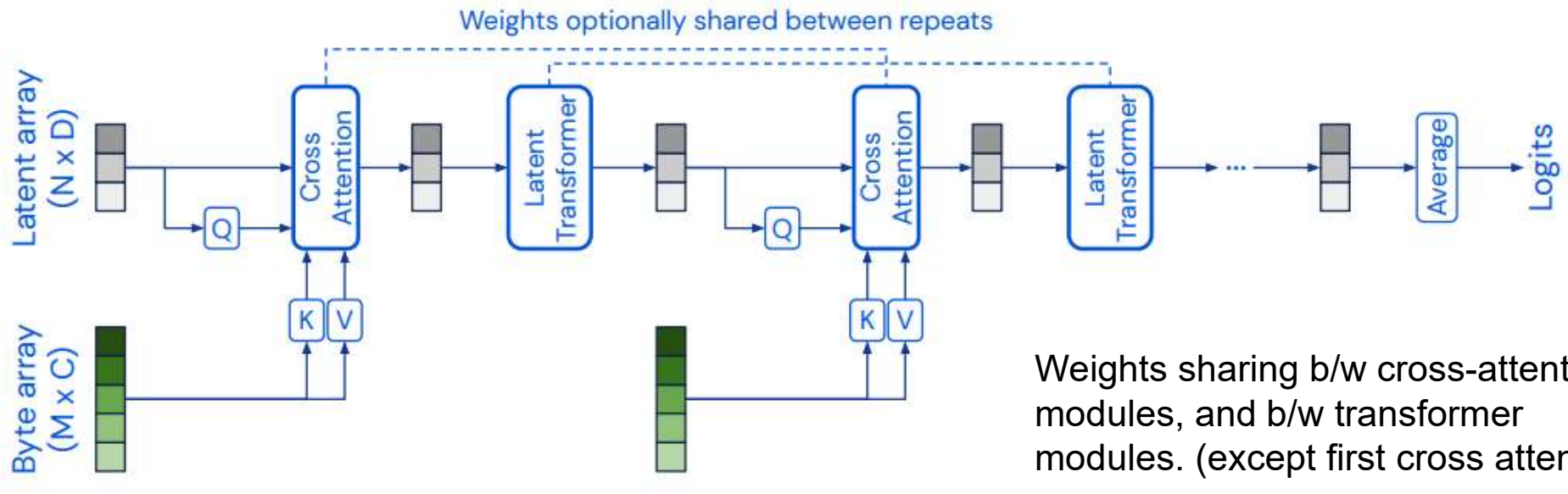
$X \in R^{n \times d}$,
$I \in R^{m \times d}$

Inducing points I are part of the ISAB itself, and they are trainable parameters.
1) The ISAB first transforms I into H by attending to the input set.

2) The set of transformed inducing points H, which Set Transformer contains information about the input set X, is again attended to by the input set X to finally produce a set of n elements

Overview

- Cross-attention module that maps a byte array and a latent array to a latent array
  > Projecting the higher-dimensional byte array through a lower-dimensional attention bottleneck.
- Transformer that maps latent array to latent array
  > Size of byte array is generally large while the size of the latent array(hyperparameter) is typically much smaller. **(N << M)**

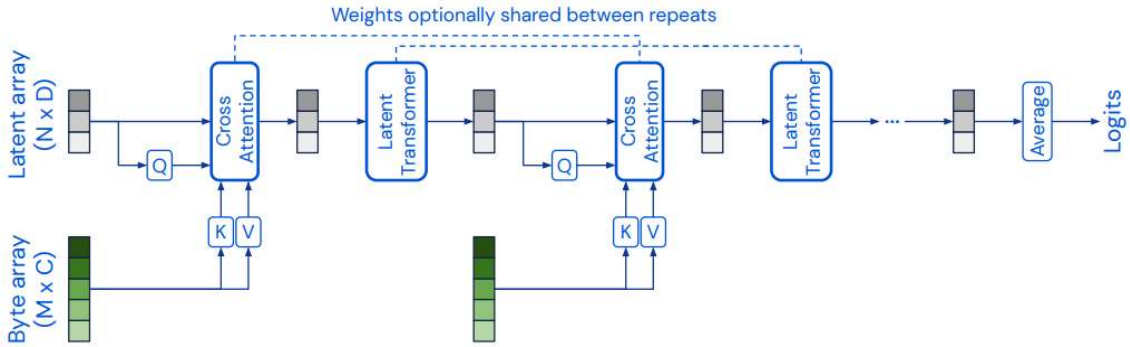


Weights sharing b/w cross-attention modules, and b/w transformer modules. (except first cross attention)

Taming Quadratic complexity with cross-attention

Attention

- Generally applicable (making less restrictive assumptions about the structure of the input data)
- Main challenge is scaling attention architectures to very large and generic inputs.



Original Transformer

$$Q = HW^Q \in R^{M \times C}$$
$$K = HW^K \in R^{M \times C}$$
$$V = HW^V \in R^{M \times C}$$

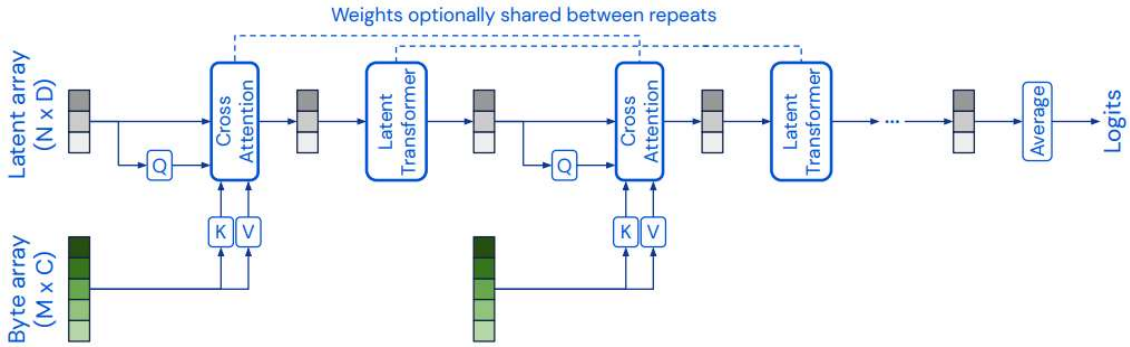
where $H \in R^{M \times C}$

Taming Quadratic complexity with cross-attention

Perceiver uses asymmetric attention

$$Perceiver$$
$$Q = ZW^Q \in R^{N \times K}$$
$$K = HW^K \in R^{M \times K}$$
$$V = HW^V \in R^{M \times K}$$

where $H \in R^{M \times C}$, $Z \in R^{N \times D}$



Cross attention and then Linear(K, D) to map latent space

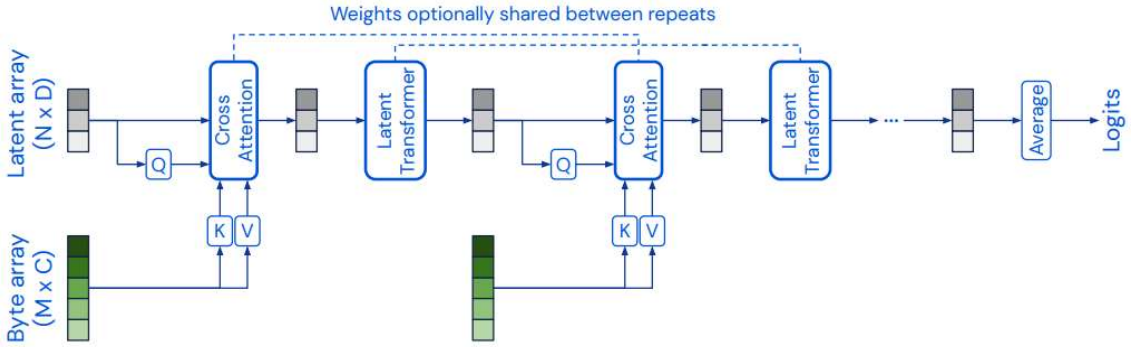
Last, Linear – GeLU - Linear

Uncoupling depth with a latent Transformer

The result of cross attention

$$\textit{Latent Transformer}$$
$$Q = ZW^Q \in R^{N\times K}$$
$$K = ZW^K \in R^{N\times K}$$
$$V = ZW^V \in R^{N\times K}$$

where Z ∈ $R^{N\times D}$ (N << M)



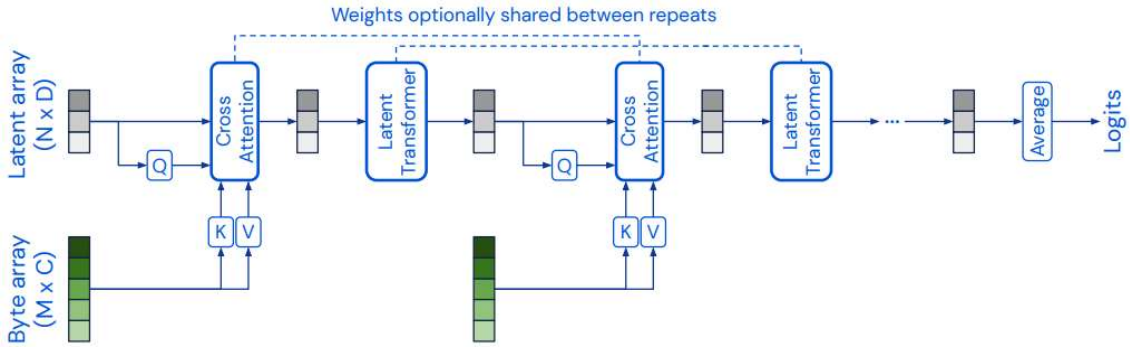
$$O(MN + LN^2)$$

This design allows Perceiver architectures to make use of much deeper Transformers.

=> Decoupling the input size and depth

Iterative cross-attention & Weight sharing

The size of latent array allows us to directly model pixels and to build deeper Transformers



But bottleneck may restrict the network's ability to capture all of the necessary details from input

Perceiver uses multiple cross-attend layers, which allow the latent array to iteratively extract information from the input.

Iterative cross-attention & Weight sharing

- Trade-off between performance and computational cost!

> More cross-attend layer means it increases the number of layers with linear dependence on the input size.

# cross-attends	Acc.	FLOPs	Params
4	39.4	173.1B	12.7M
8	45.3	346.1B	23.8M
12	OOM	519.2B	34.9M

Table 5. Performance of models built from a stack of cross-attention layers with no latent transformers. We do not share weights between cross-attention modules in this experiment. Models with 12 cross-attends run out of memory on the largest device configuration we use (64 TPUs). Results are top-1 validation accuracy (in %) on ImageNet (higher is better).

# cross-attends	Acc.	FLOPs	Params
1 (at start)	76.7	404.3B	41.1M
1 (interleaved)	76.7	404.3B	42.1M
2 (at start)	76.7	447.6B	44.9M
2 (interleaved)	76.5	447.6B	44.9M
4 (at start)	75.9	534.1B	44.9M
4 (interleaved)	76.5	534.1B	44.9M
8 (at start)	73.7	707.2B	44.9M
8 (interleaved)	**78.0**	707.2B	44.9M

Table 6. Performance as a function of # of cross-attends and their arrangement. In "interleaved," cross-attention layers are spaced throughout the network (for re-entrant processing), while in "at start" all cross-attends are placed at the start of the network followed by all latent self-attend layers. All cross-attention layers except the initial one are shared, and self-attends are shared as usual (using 8 blocks of 6 self-attention modules). Results are top-1 validation accuracy (in %) on ImageNet (higher is better).

Iterative cross attention & Weight sharing

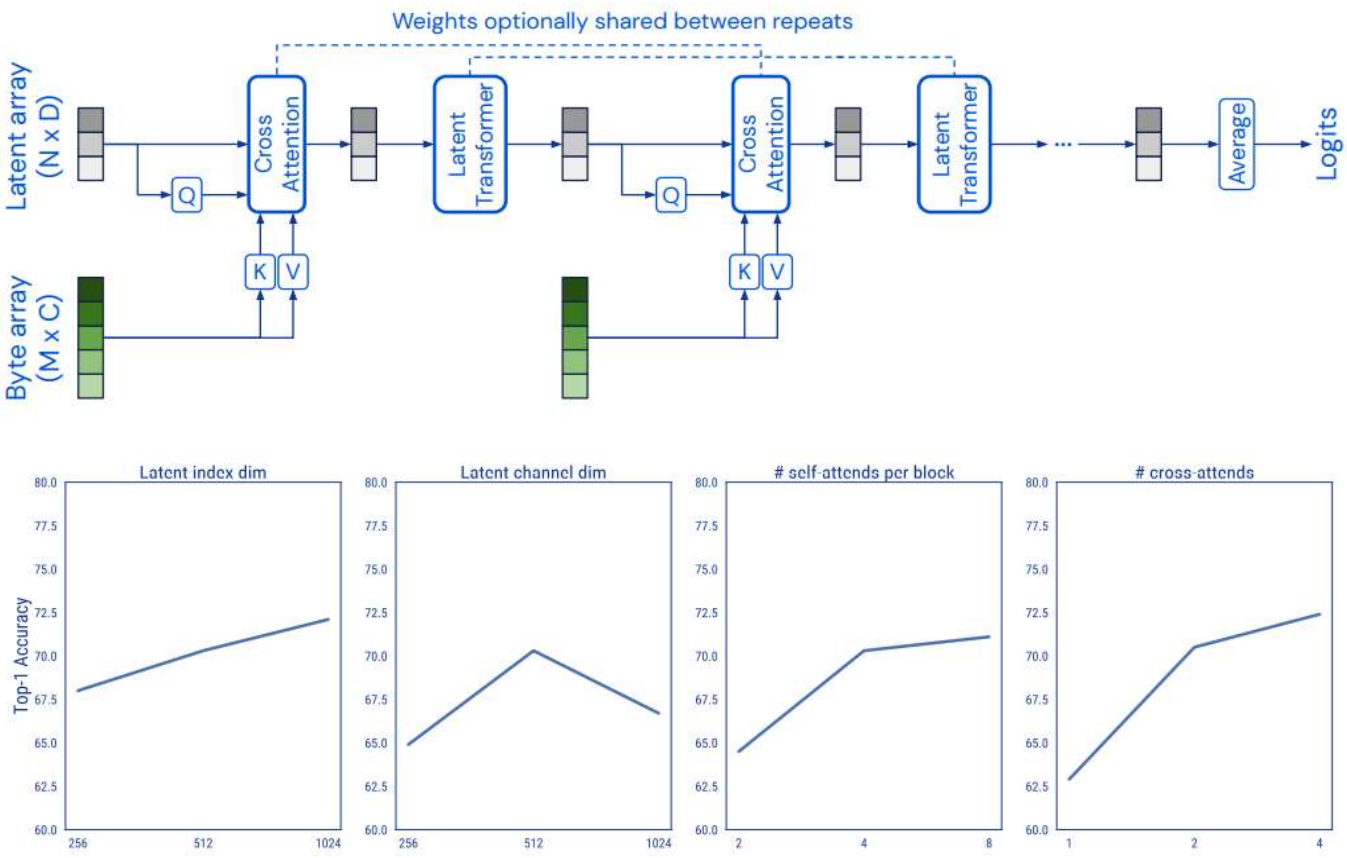


Figure 5. The effect of model hyperparameters, using a scaled-down Perceiver architecture on ImageNet. All plots show top-1 accuracy (higher is better). Increasing the size of the latent index dimension, the number of self-attends per block, and the number of cross-attends generally produced better results. Increasing the size of the latent channel dimension helps up to a point, but often leads to overfitting.

Positional Encodings

A pure attention model will return the same output regardless of the order of its inputs (Permutation Invariance)

This property makes attention-based architectures well-suited for many types of data, as they make no assumptions about which spatial relationships or symmetries to prioritize.

But permutation invariance means that the Perceiver's architecture cannot in and of itself exploit spatial relationships in the input data.

position information is typically injected by tagging position encodings onto the input features

Positional Encodings

- Use Scalable Fourier features (same as Transformer)

> directly represent the position structure of the input data (preserving 1D temporal or 2D spatial structure for audio or images, respectively, or 3D spatiotemporal structure for videos)

Transformer

$$\mathrm{PE}(pos, 2i) = sin\left(\frac{pos}{10000^{2i/d_{model}}}\right),$$

$$\mathrm{PE}(pos, 2i+1) = cos\left(\frac{pos}{10000^{2i/d_{model}}}\right).$$

For example, for word w at position $pos \in [0, L-1]$ in the input sequence $\boldsymbol{w} = (w_0, \cdots, w_{L-1})$, with 4-dimensional embedding e_w, and $d_{model} = 4$, the operation would be

$$e'_w = e_w + \left[sin\left(\frac{pos}{10000^0}\right), cos\left(\frac{pos}{10000^0}\right), sin\left(\frac{pos}{10000^{2/4}}\right), cos\left(\frac{pos}{10000^{2/4}}\right)\right]$$
$$= e_w + \left[sin\left(pos\right), cos\left(pos\right), sin\left(\frac{pos}{100}\right), cos\left(\frac{pos}{100}\right)\right]$$

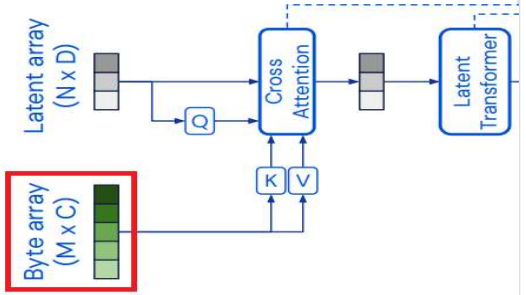
Perceiver

$$[\sin(f_k \pi x_d), \cos(f_k \pi x_d)]$$

Where the frequency k is the k^{th} band of bank of frequencies, and x_d is the value of the input position along the d^{th} dimension (normalize to [-1, 1])

Positional Encodings

Does position encodings undermine generality? NO
Feature based approach allows network to learn how to use the position structure.
Also, P.E can be easily adapted to new domain



Perceiver

$$[\sin(f_k \pi x_d), \cos(f_k \pi x_d)]$$

Finally, we concatenate the raw position value x_d which leads to position encoding of size d(2K+1)

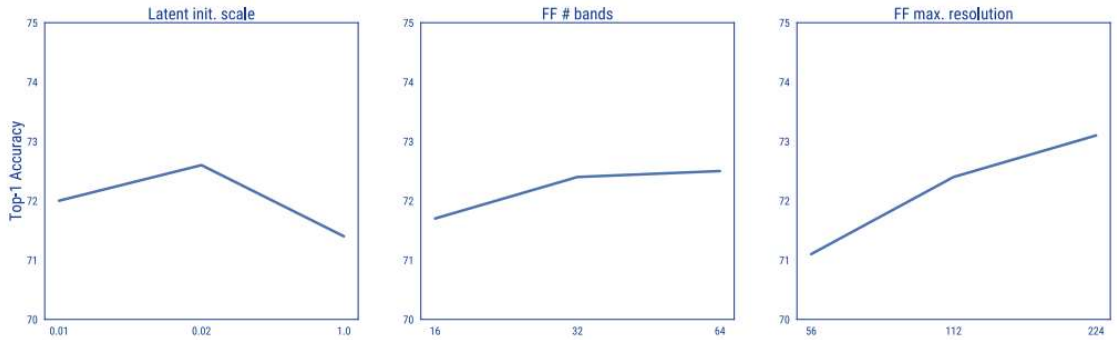


Figure 6. The effect of latent initialization scale and Fourier feature (FF) position encoding parameters on performance. All plots show top-1 accuracy (higher is better). The model with initialization scale of 0.1 diverged during training. Generally, increasing the number of bands and max resolution (up to Nyquist) increased performance. We observed the same effects whether using linearly or logarithmically spaced position encoding bands.

More

	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

Table 2. Top-1 validation accuracy (in %) on standard (raw) and **permuted** ImageNet (higher is better). Position encodings (in parentheses) are constructed before permutation, see text for details. While models that only use global attention are stable under permutation, models that use 2D convolutions to process local neighborhoods are not. The size of the local neighborhood at input is given by the input receptive field (RF) size, in pixels.

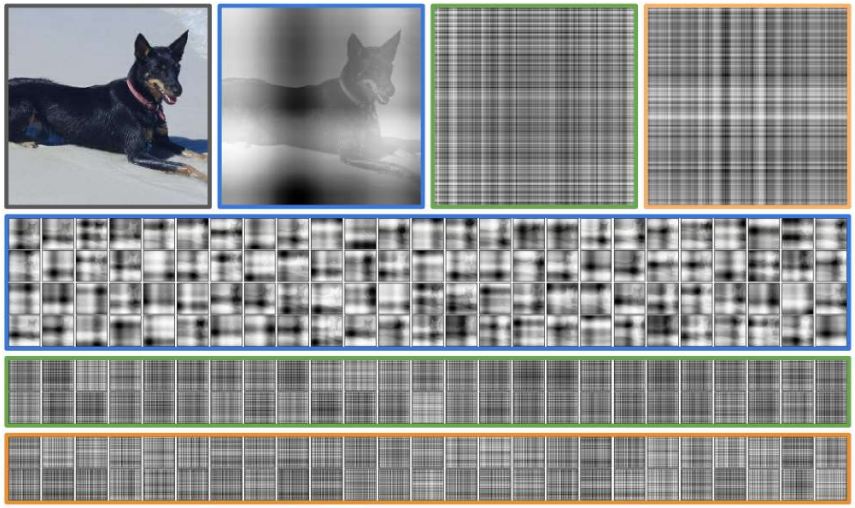


Figure 3. Attention maps from the **first**, **second**, and **eighth** (final) cross-attention layers of a model on ImageNet with 8 cross-attention modules. Cross-attention modules 2-8 share weights in this model. **Row 1:** Original image and close-ups of one attention map from each of these layers. **Rows 2-4:** Overview of the attention maps of the cross-attention modules. Attention maps appear to scan the input image using tartan-like patterns at a range of spatial frequencies. The visualized attention maps are *not* overlaid on the input image: any apparent image structure is present in the attention map itself (the dog is clearly visible in several of the first module's attention maps).

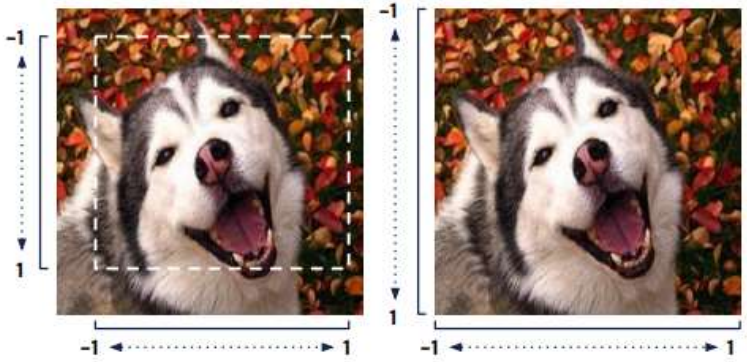
Experiments

- Images - ImageNet

  1) Preprocessing – Inception-style processing,

     and RandAugment (Ekin D. Cubuk) which is modality specific

  2) Position encodings – crop and then do P.E



(a) Crop-relative coordinates (b) Image-relative coordinates

Figure 4. For ImageNet experiments, we generate position encodings using [-1, 1]-normalized (x, y)-coordinates drawn from (a) crops rather than from (b) the raw images, as we find the latter leads to overfitting.

ResNet-50 (He et al., 2016)	77.6
ViT-B-16 (Dosovitskiy et al., 2021)	77.9
ResNet-50 (FF)	73.5
ViT-B-16 (FF)	76.7
Transformer (64x64, FF)	57.0
Perceiver (FF)	78.0

Table 1. Top-1 validation accuracy (in %) on ImageNet. Models that use 2D convolutions exploit domain-specific grid structure architecturally, while models that only use global attention do not.

Others

	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**

Table 4. Top-1 test-set classification accuracy (in %) on ModelNet40. Higher is better. We report best result per model class, selected by test-set score. There are no RGB features nor a natural grid structure on this dataset. We compare to the generic baselines considered in previous sections with Fourier feature encodings of positions, as well as to a specialized model: PointNet++ (Qi et al., 2017). PointNet++ uses extra geometric features and performs more advanced augmentations that we did not consider here and are not used for the models in blue.

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

Table 3. Perceiver performance on AudioSet, compared to state-of-the-art models (mAP, higher is better).

Figure 2. We train the Perceiver architecture on images from ImageNet (Deng et al., 2009) (left), video and audio from AudioSet (Gemmeke et al., 2017) (considered both multi- and uni-modally) (center), and 3D point clouds from ModelNet40 (Wu et al., 2015) (right). Essentially no architectural changes are required to use the model on a diverse range of input data.