INFO NOVELTY INTERESTING

Perceiver: General Perception with Iterative Attention

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

arXiv:2103.03206v2 [cs.CV] 23 Jun 2021

Abstract

Biological systems perceive the world by simultaneously processing high-dimensional inputs from modalities as diverse as vision, audition, touch, proprioception, etc. The perception models used in deep learning on the other hand are designed for individual modalities, often relying on domain-specific assumptions such as the local grid structures exploited by virtually all existing vision models. These priors introduce helpful inductive biases, but also lock models to individual modalities. In this paper we introduce *the Perceiver* – a model that builds upon Transformers and hence makes few architectural assumptions about the relationship between its inputs, but that also scales to hundreds of thousands of inputs, like ConvNets. The model leverages an asymmetric attention mechanism to iteratively distill inputs into a tight latent bottleneck, allowing it to scale to handle very large inputs. We show that this architecture is competitive with or outperforms strong, specialized models on classification tasks across various modalities: images, point clouds, audio, video, and video+audio. The Perceiver obtains performance comparable to ResNet-50 and ViT on ImageNet without 2D convolutions by directly attending to 50,000 pixels. It is also competitive in all modalities in AudioSet.

1. Introduction

Inductive biases such as spatial locality in early vision are clearly valuable and are famous for drastically increasing the efficiency of learning perceptual models. 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? Or are we better off building in as much flexibility as possible, and encouraging the data to speak for itself (LeCun et al., 2015)?

One glaring issue with strong architectural priors is that they are often modality-specific. For example, if we assume that the input is a single image, we can use our knowledge of its 2D grid structure and build an efficient architecture that relies on 2D convolutional operations. But if we move to a stereo pair, we must decide how to modify this structure to jointly process the pixels from both sensors: should we use an early or late fusion architecture (Karpathy et al., 2014) or should we sum or concatenate features? If we move to audio, then the merits of a 2D grid are no longer as clear, and a different type of model, such as 1D convolutions or an LSTM (Hochreiter & Schmidhuber, 1997; Graves et al., 2013), may be warranted instead. If we want to process point clouds – a common concern for self-driving cars equippped with Lidar sensors – then we can longer rely on models that scale best for fixed, low-resolution grids. In short, using standard tools, we are forced to redesign the architecture we use every time the input changes.

In this paper we introduce the *Perceiver*, a model designed to handle arbitrary configurations of different modalities using a single Transformer-based architecture. Transformers (Vaswani et al., 2017) are very flexible architectural blocks that make few assumptions about their inputs, but that also scale quadratically with the number of inputs, in terms of both memory and computation. Recent work has shown impressive performance using Transformers on images, but this work relies on the pixels' grid structure to reduce computational complexity, by first processing pixels using a 2D convolution (Dosovitskiy et al., 2021; Touvron et al., 2020), factorizing the image into columns and rows (Ho et al., 2019; Child et al., 2019), or by aggressive subsampling (Chen et al., 2020a). Instead, we propose a mechanism that can handle high-dimensional inputs while retaining the expressivity and flexibility needed to deal with arbitrary input configurations.

Our core idea is to introduce a small set of latent units that forms an attention bottleneck through which the inputs must pass (Fig. 1). This eliminates the quadratic scaling problem of all-to-all attention of a classical Transformer and decouples the network depth from the input's size, allowing us to construct very deep models. By attending to the inputs iteratively, the Perceiver can channel its limited capacity to the most relevant inputs, informed by previous steps. But spatial or temporal information is crucial for many modali-

[1]DeepMind – London, UK. Correspondence to: Andrew Jaegle <drewjaegle@deepmind.com>.

Proceedings of the 38^{th} International Conference on Machine Learning, PMLR 139, 2021. Copyright 2021 by the author(s).

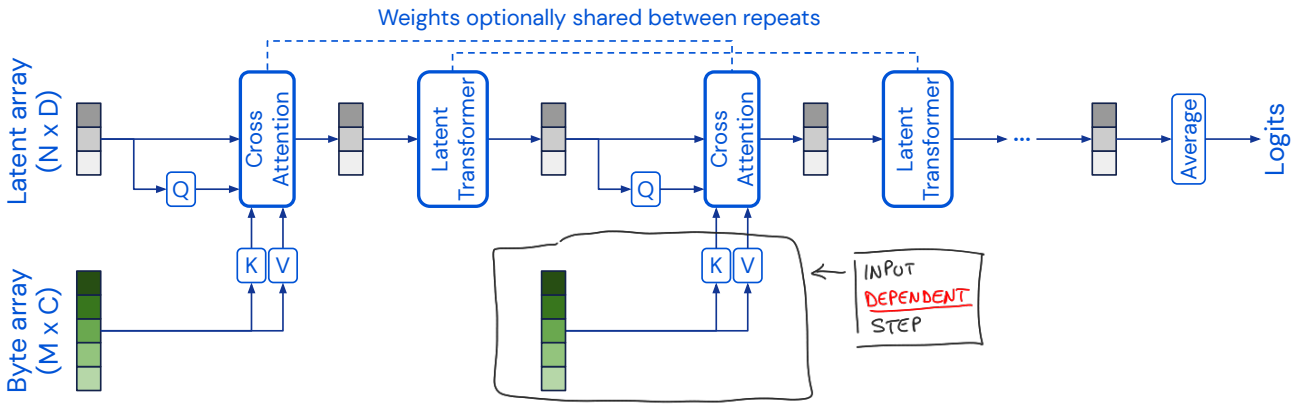


Figure 1. The Perceiver is an architecture based on attentional principles that scales to high-dimensional inputs such as images, videos, audio, point-clouds, and multimodal combinations without making domain-specific assumptions. The Perceiver uses a cross-attention module to project an high-dimensional input byte array to a fixed-dimensional latent bottleneck (the number of input indices M is much larger than the number of latent indices N) before processing it using a deep stack of Transformer-style self-attention blocks in the latent space. The Perceiver iteratively attends to the input byte array by alternating cross-attention and latent self-attention blocks.

ties, and it is often essential to distinguish input from one modality or another in multimodal contexts. We can compensate for the lack of explicit structures in our architecture by associating position and modality-specific features with every input element (e.g. every pixel, or each audio sample) – these can be learned or constructed using high-fidelity Fourier features (Mildenhall et al., 2020; Tancik et al., 2020; Vaswani et al., 2017). This is a way of tagging input units with a high-fidelity representation of position and modality, similar to the labeled lined strategy used to construct topographic and cross-sensory maps in biological neural networks by associating the activity of a specific unit with a semantic or spatial location (Kandel et al. 2012, Ch. 21).

We demonstrate performance comparable to strong models such as ResNet-50 and ViT when training on ImageNet for classification; competitive performance on the AudioSet sound event classification benchmark (using raw audio, video, or both); and strong performance relative to comparable approaches on ModelNet-40 point cloud classification.

2. Related Work

ConvNets (Fukushima, 1980; LeCun et al., 1998; Cireşan et al., 2011; Krizhevsky et al., 2012) have been the dominant family of architectures for perceptual tasks for nearly a full decade, thanks to their good performance and scalability. They can handle high-resolution images while using relatively few parameters and relatively little compute by using convolutions to share weights in 2D and limit each unit's computation to a local 2D neighborhood. However, as discussed in the previous section, they offer limited flexibility when combining multiple signals, unlike the attention-based models dominant in language, as exemplified by Transformers (Vaswani et al., 2017).

Efficient attention architectures. Transformers are amazingly flexible but scale poorly with the input size because all self-attention layers have the same number of inputs and standard self-attention compares each input to every other input at all layers. Nevertheless, self-attention has been rapidly percolating into perception, for example as pieces of otherwise convolutional models for images (Bello et al., 2019; Cordonnier et al., 2020; Srinivas et al., 2021) and videos (Wang et al., 2018; Girdhar et al., 2019). A variety of strategies have been proposed to reduce the size of the input to the Transformer so it can be used on domains that are otherwise too large, including subsampling the input (Chen et al., 2020a) or by first preprocessing the input using convolutions (e.g. Wu et al. 2020). This is the strategy taken by the Vision Transformer (ViT) (Dosovitskiy et al., 2021), which first reduces the input size to ~ 200 using a 2D convolutional layer (referred to as "linear projection of flattened patches" in that work) and then applying a Transformer on the resulting inputs along with a BERT-like class token (Devlin et al., 2019). ViT produces impressive results on ImageNet but this preprocessing strategy restricts it to image-like domains with grid-like sampling patterns.

Several groups have proposed to modify the internals of the Transformer's self-attention module to gain greater efficiency (see Appendix Sec. A for a discussion). Most closely related to our work is the Set Transformer (Lee et al., 2019). The Set Transformer uses cross-attention to project a large input array to a smaller array, either to reduce the computation within a module or to project inputs to a target output shape (e.g. mapping an input set to logits). Like this work, the Perceiver uses cross-attention over an auxiliary low-dimensional array to reduce the complexity of attention from quadratic to linear in the input size. In a similar vein (but without using cross-attention), Linformer (Wang et al.,



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.

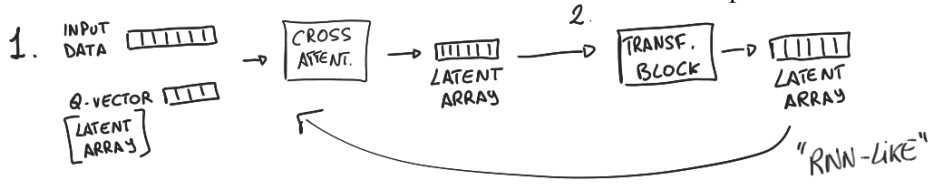
2020b) produces linear-complexity self-attention modules by projecting key and value inputs to arrays with a size smaller than the input. Unlike this prior work, the Perceiver uses cross-attention not only to get linear complexity layers, but also to decouple network depth from the input size. As discussed in Sec. 3, it is this decoupling and not merely linear scaling that allows us to build very deep architectures, which appear to be essential for good performance on challenging tasks in a range of domains. We discuss the relationship between the Perceiver and the Set Transformer and related models in more detail in Appendix Sec. A.

Multimodal architectures. In current approaches to multi-modal processing, separate feature extractors are used for each modality (Kaiser et al., 2017; Arandjelovic & Zisserman, 2018; Wang et al., 2020c; Chen et al., 2020b; Alayrac et al., 2020; Lee et al., 2020; Xiao et al., 2020) – it is generally not sensible to concatenate an audio spectrogram or a raw audio waveform with an image and pass it through a ConvNet. This approach leads to a variety of architectural choices – such as when to fuse modalities – that need to be re-tuned for each application. Because of this state of affairs, best-practice architectures for vision cannot be ported to all domains, and specialized models have been developed to handle domains like point clouds (Qi et al., 2017; Guo et al., 2020). The Perceiver is designed to very flexibly handle a wide range of inputs out of the box even if they come from very different modalities, including high-bandwidth ones such as images and audio (as illustrated by Fig. 2).

3. Methods

3.1. The Perceiver architecture

Overview. We build our architecture from two components: (i) a cross-attention module that maps a byte array (e.g. an pixel array) and a latent array to a latent array, and (ii) a Transformer tower that maps a latent array to a latent array. The size of the byte array is determined by the input data and is generally large (e.g. ImageNet images at resolution 224 have 50,176 pixels), while the size of the latent array is a hyperparameter which is typically much smaller (e.g. we use 512 latents on ImageNet). Our model applies the cross-attention module and the Transformer in alternation. This corresponds to projecting the higher-dimensional byte array through a lower-dimension attention bottleneck before processing it with a deep Transformer, and then using the resulting representation to query the input again. The model can also be seen as performing a fully end-to-end clustering of the inputs with latent positions as cluster centres, leveraging a highly asymmetric cross-attention layer. Because we optionally share weights between each instance of the Transformer tower (and between all instances of the cross-attention module but the first), our model can be interpreted as a recurrent neural network (RNN), but unrolled in depth using the same input, rather than in time. All attention modules in the Perceiver are non-causal: we use no masks. The Perceiver architecture is illustrated in Fig. 1.



Taming quadratic complexity with cross-attention. We structure our architecture around attention because it is both generally applicable (making less restrictive assumptions about the structure of the input data than e.g. ConvNets; it's all you need) and powerful in practice. The main challenge addressed by our architecture's design is scaling attention architectures to very large and generic inputs. Both cross-attention and Transformer modules are structured around the use of query-key-value (QKV) attention (Graves et al., 2014; Weston et al., 2015; Bahdanau et al., 2015). QKV attention applies three networks – the query, key, and value networks, which are typically multi-layer perceptrons (MLPs) – to each element of an input array, producing three arrays that preserve the index dimensionality (or *sequence length*) M of their inputs. The main difficulty of using Transformers on large-scale inputs like images is that the complexity of QKV self-attention is quadratic in the input index dimensionality, but the index dimensionality M of images is typically very large ($M = 50176$ for 224×224 ImageNet images). The challenge is similar for audio: 1 second of audio at standard sampling rates corresponds to around 50,000 raw audio samples. This problem compounds dramatically for multi-modal inputs.

For this reason, prior work that uses attention to process images avoids directly applying standard QKV attention to the input pixel array (see Sec. 2 and Appendix Sec. A for an overview). Here, we apply attention directly to the inputs by introducing an asymmetry into the attention operation. To see how this works, first note that for $Q \in \mathbb{R}^{M \times D}$, $K \in \mathbb{R}^{M \times C}$, and $V \in \mathbb{R}^{M \times C}$, (where C and D are channel dimensions) the complexity of the QKV attention operation – essentially, $\text{softmax}(QK^T)V$ – is $\mathcal{O}(M^2)$, as it involves two matrix multiplications with matrices of large dimension M.[1] So we introduce asymmetry: while K and V are projections of the input byte array, Q is a projection of a learned latent array with index dimension $N \ll M$, where the latent's index dimension N is a hyperparameter. The resulting cross-attention operation has complexity $\mathcal{O}(MN)$.

KEY POINT N << M $N^2 \ll M^2$

Uncoupling depth with a latent Transformer. The output of the cross-attention module takes the shape of the input to the Q network: that is, the cross-attention layer induces a bottleneck. We exploit this bottleneck by building deep (and hence expressive) Transformers in the latent space: they come at the low cost of $\mathcal{O}(N^2)$. This design allows Perceiver-based architectures to make use of much deeper Transformers than efficient Transformers that use linear-complexity layers, without relying on domain-specific assumptions. This is because a Transformer built on bytes has complexity $\mathcal{O}(LM^2)$ while a latent Transformer has complexity $\mathcal{O}(LN^2)$ (where $N \ll M$), when considered as a function of the number of layers L in addition to index dimensionality.

This results in an architecture with complexity $\mathcal{O}(MN + LN^2)$, and this is key: by decoupling the input size and the depth, we can add additional Transformer layers at a cost that's independent of the input size. This allows us to construct very large networks on large-scale data. For example, our best ImageNet results use a network with 48 latent Transformer blocks, which is infeasible with networks that couple input size and depth (e.g. see Tab. 5).

Our latent Transformer uses the GPT-2 architecture (Radford et al., 2019), which itself is based on the decoder of the original Transformer architecture (Vaswani et al., 2017). In our experiments, we use values of $N \leq 1024$, which makes our latent Transformer comparable in input size to models in wide-spread use in language. The latent array itself is initialized using a learned position encoding (Gehring et al., 2017) (see Appendix Sec. C for details).

Iterative cross-attention & weight sharing. The size of the latent array allows us to directly model pixels and to build deeper Transformers, but the severity of the bottleneck may restrict the network's ability to capture all of the necessary details from the input signal. To hedge against this effect, the Perceiver may be structured with multiple cross-attend layers, which allow the latent array to iteratively extract information from the input image as it is needed. This allows us to tune the model to balance expensive, but informative cross-attends against cheaper, but potentially redundant latent self-attends. As shown in Appendix Tab. 6, more cross-attends leads to better performance, but increases the computational requirements of the model because it increases the number of layers with linear dependence on the input size.

"NOT REALLY INDEPENDENT"

Finally, in virtue of the iterative structure of the resulting architecture, we can increase the parameter efficiency of the model by sharing weights between the corresponding blocks of each latent Transformer and/or between cross-attend modules. Latent self-attention blocks can still be shared if only a single cross-attend is used. In our ImageNet experiments, weight sharing results in an approximately 10x reduction in the number of parameters, while reducing overfitting and boosting validation performance. The resulting architecture has the functional form of an RNN with a cross-attention input projection, a bottlenecked latent dimensionality, and a latent Transformer recurrent core. We note that weight sharing has been used for similar goals in Transformers (Dehghani et al., 2019; Lan et al., 2020).

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. The first block reports standard performance from pixels – these numbers are taken from the literature. The second block shows performance when the inputs are RGB values concatenated with 2D Fourier features (FF) – the same that the Perceiver receives. This block uses our implementation of the baselines. The Perceiver is competitive with standard baselines on ImageNet without relying on domain-specific architectural assumptions.

3.2. Position encodings

Permutation invariance and position information. Attention is a permutation-invariant operation, and this property is preserved by the Perceiver and related models (Lee et al., 2019). A pure attention model will return the same output regardless of the order of its inputs, leaving no trace of the input's ordering on its outputs. 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. In contrast, the

[1] We ignore the contributions of the channel dimensions C and D here, as they are generally small relative to M.

	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.

ConvNets that are typically used in image processing – such as residual networks (ResNets) (He et al., 2016) – bake in 2D spatial structure in several ways, including by using filters that look only at local regions of space (which makes it easier to capture the relationship between nearby pixels than between distant pixels), by sharing weights across both spatial dimensions (which helps to model data with statistics that are invariant to translation), and by repeatedly applying small filters (which helps to model data with statistics that are invariant to scale).

But permutation invariance means that the Perceiver's architecture cannot in and of itself exploit spatial relationships in the input data. Spatial relationships are essential for sensory reasoning (Kant, 1781) and this limitation is clearly unsatisfying. In the attention literature, position information is typically injected by tagging *position encodings* onto the input features (Vaswani et al., 2017); we pursue this strategy here as well. While position information is typically used to encode sequence position in the context of language, it can also be used to encode spatial, temporal, and modality identity.

Scalable Fourier features. Here, we use a strategy that has recently gained renewed prominence, both in language and in vision: Fourier feature position encodings (Stanley, 2007; Vaswani et al., 2017; Parmar et al., 2018; Tancik et al., 2020; Mildenhall et al., 2020). We use a parameterization of Fourier features that allows us to (i) 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), (ii) control the number of frequency bands in our position encoding independently of the cutoff frequency, and (iii) uniformly sample all frequencies up to a target resolution.

We parametrize the frequency encoding to take the values $[\sin(f_k \pi x_d), \cos(f_k \pi x_d)]$, where the frequency f_k is the k^{th} band of a bank of frequencies spaced equally between 1 and $\frac{\mu}{2}$. $\frac{\mu}{2}$ can be naturally interpreted as the Nyquist frequency (Nyquist, 1928) corresponding to a target sampling rate of μ. By allowing the network to resolve all positions in an input array, we can encourage it to learn to compare the values of bytes at any positions in the input array. x_d is the value of the input position along the d^{th} dimension (e.g. for images $d = 2$ and for video $d = 3$). x_d takes values in $[-1, 1]$ for each dimension. We concatenate the raw position value x_d to produce the final representation of position. This results in a position encoding of size $d(2K + 1)$.

This parameterization is related to the NeRF position encoding scheme (Mildenhall et al., 2020), which is built around frequency bands with increasing powers of two (the k^{th} band has frequency 2^k). This leads to very high frequencies for even modest numbers of bands, and in some experiments, we encountered numerical instability when using this parameterization beyond around $k = 15$ bands.

In language modelling, Transformer inputs are typically produced by adding a position encoding to the input encoding (the size of the position encoding is tailored to the encoding used). We found it beneficial to instead concatenate the position and input features before passing them into the Perceiver. This difference is perhaps explained by the fact that input features in language tend to be larger and sparser than the modalities considered here.

I THINK THEY TESTED BOTH!

Position encodings are generally applicable. Does the use of position encodings undermine our claim to be moving from a more domain-specific architecture built to exploit 2D structure to a more general ones? No, for three reasons. **First,** while the architectural imposition of position information hard codes a specific positional prior, the feature-based approach allows the network to learn how to use (or ignore) the position structure. This is in accord with the idea that greater generality follows from making as much of a system learnable as possible (Sutton, 2019). **Second,** it is possible to redesign architectural priors for data domains with different structures, such as videos (Tran et al., 2015) or audio (Ford et al., 2019), or for groups other than the group of linear translations (e.g. Cohen & Welling 2016; Bronstein et al. 2017; Esteves et al. 2018)); this however often requires a tremendous amount of researcher time and expertise. In contrast, a position encoding can be easily adapted to a new domain: Fourier features are trivial to adapt as long as the input dimensionality is relatively small and known. In the broader Transformer literature, simple learned position encoding have proven to be sufficient for good results in many settings. We find that a similar strategy produces reasonable results on ImageNet (see Table 2, bottom row) even though it has no knowledge whatsoever about the input 2D structure. **Third,** position encodings can be naturally extended to multimodal data: each domain can use a position encoding with the correct dimensionality for its data, with learned encodings used to distinguish domains (we use this strategy for multimodal Audioset, see Sec. 4.2).

K WELL...

THIS IS THE MOST CONVINCING REASON!

4. Experiments

The next few subsections are organized by the modalit{y, ies} used (illustrated in Fig. 2). We evaluate the effect of model configuration and hyperparameters on ImageNet classification in the supplement (Sec. B). As baselines we consider ResNet-50 (He et al., 2016), a very widely model for both vision and audio and possibly the closest thing to a general perceptual architecture so far. We also consider two Transformer variants, the recently proposed ViT (Dosovitskiy et al., 2021), and a stack of Transformers (Vaswani et al., 2017). All experiments were conducted using JAX (Bradbury et al., 2018) and the DeepMind JAX ecosystem (Babuschkin et al., 2020).

4.1. Images – ImageNet

First, we consider the task of single-image classification using the ILSVRC 2012 split of the ImageNet dataset (Deng et al., 2009). ImageNet has been a crucial bellwether in the development of architectures for image recognition (Krizhevsky et al., 2012; Simonyan & Zisserman, 2015; Szegedy et al., 2015; He et al., 2016) and, until recently, it has been dominated by ConvNet architectures. Each image on ImageNet has a single label so we use softmax outputs and a cross-entropy loss to train for the classification task. As is standard practice, we evaluate our model and all baselines using the top-1 accuracy on the held-out validation set (the test set is not publicly available). We train our model using images sampled by Inception-style preprocessing (Szegedy et al., 2015), including standard 224×224 pixel crops. Additionally, we augment all images using RandAugment (Cubuk et al., 2020) at training time.

Position encodings. We generate position encodings by first using the (x, y) positions on the 224×224 input crop. (x, y) coordinates are standardized to [-1, 1] for each dimension of a crop (see Appendix Fig. 4). In Inception-style preprocessing, the raw crop can have a non-uniform aspect ratio, which may lead to aspect ratio distortion in both the input crop and in the (x, y) coordinates used to generate the position encoding. In early experiments, we tried using image coordinates rather than crop coordinates as the basis of the position encoding, but we found that this led to model overfitting. We suspect that this occurs because the Perceiver's architecture may allow it to memorize training-set image by latching onto a small number of input pixels, if they are always associated with the same (RGB, position) feature. By using crops, we effectively introduce augmentation in both position and aspect ratio, which breaks correlations between RGB values and position features and makes it much harder to associate an image label with a small number of pixels.

Optimization and hyperparameters. Although it is typical to train convolutional networks on ImageNet using SGD, we found it easier to optimize Perceiver models using the LAMB optimizer (You et al., 2020), which was developed for optimizing Transformer-based models. We trained models for 120 epochs with an initial learning rate of 0.004, decaying it by a factor of 10 at [84, 102, 114] epochs. The best-performing Perceiver we identified on ImageNet attends to the input image 8 times, each time processing the full 50,176-pixel input array using a cross-attend module and a latent Transformer with 6 blocks and one cross-attend module with a single head per block. We found that sharing the initial cross-attention with subsequent cross-attends led to instability in training, so we share all cross-attends after the first. The dense subblock of each Transformer block doesn't use a bottleneck. We used a latent array with 512 indices and 1024 channels, and position encodings generated with 64 bands and a maximum resolution of 224 pixels. On ImageNet, we found that models of this size overfit without weight sharing, so we use a model that shares weights for all but the first cross-attend and latent Transformer modules. The resulting model has ~ 45 million parameters, making it comparable in size to convolutional models used on ImageNet.

Standard ImageNet. As shown in Table 1, the Perceiver model we trained on ImageNet obtains results that are competitive with models specifically designed for processing images. We include ResNet-50 results from (Cubuk et al., 2020), as these numbers use RandAugment and hence better match our training protocol. To account for the Perceiver's use of Fourier features at input, we trained versions of the benchmark models with this input as well and found that it produced comparable, if slightly worse, performance to models trained solely on RGB input. Additionally, we tested the performance of a pure Transformer model. Because Transformers cannot handle ImageNet-scale data, we first downsampled the input images to 64×64 before passing it into the Transformer (we obtained similar results using 96x96 inputs, which however is much slower to train and more memory-intensive so we could not use as many layers). The Transformer model we consider has the same architecture as the latent Transformer of the Perceiver, differing only in hyperparameters (we swept each model independently), for more details please consult the Appendix. Note that state-of-the art on ImageNet without pretraining was 86.5% top-1 accuracy at submission (Brock et al., 2021).

Permuted ImageNet. To evaluate how important domain-specific assumptions about grid structure are to the performance of the benchmark methods, we evaluate all methods on permuted ImageNet. To generate permutations, we use a single, shared permutation pattern for all images. The permutation is performed *after* position features are generated. We make this choice because it still allows each network to infer the spatial relationship between points (using the position encoding), but prevents the network from using an

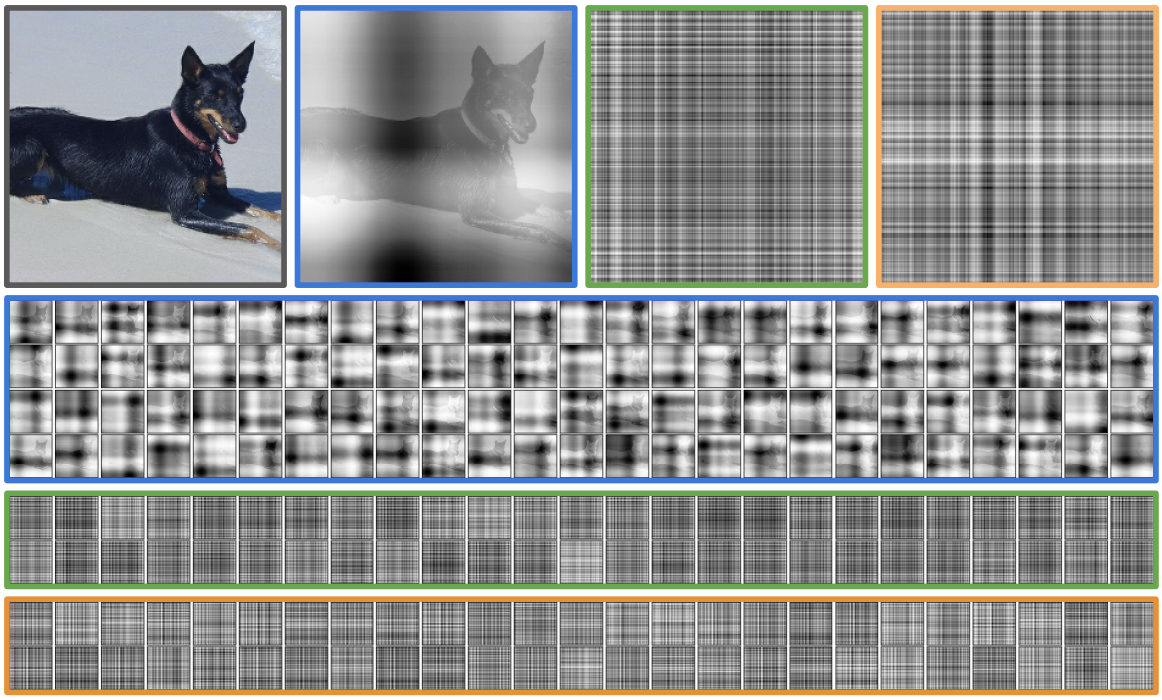


Figure 3. Attention maps from the **first**, **second**, and **eighth** (final) cross-attention layers of a model on ImageNet with 8 cross-attention modules. Cross-attention modules 2-8 share weights in this model. **Row 1:** Original image and close-ups of one attention map from each of these layers. **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).

architectural inductive bias to do so. In other words, under these conditions, networks that use 2D convolutions cannot exploit the local neighorhood structure of inputs to reason about space, but must reason in terms of features, just like structure-agnostic architectures like Transformers and Perceivers. The results of this experiment are shown in Table 2. As the Transformer and Perceiver effectively treat any input as a permuted input, their results are not affected, but we find that the performance of both ViT and ResNet suffer dramatically after permutation, even though these models have access to position encodings. Both of these models still perform far above chance, suggesting they are able to reason in terms of position features. ViT is more stable under permutation than the ResNet model: this is likely because ViT uses a single 2D convolution layer with a fairly large receptive field ($16 \times 16 = 256$ pixels) followed by a Transformer, while ResNet-50 uses an initial convolution with a relatively small receptive field ($7 \times 7 = 49$ pixels) and an architecture essentially entirely composed of convolutions.

The Perceiver architecture itself makes no assumptions about the spatial structure of its input, but the Fourier features position encoding we use by default does. By replacing these features with a fully learned, 128-dimensional position encoding, we can evaluate the performance of a Perceiver **with no knowledge of the spatial structure of the inputs**. The results of this experiment are shown in the bottom row of Table 2. The position encoding used here is initialized randomly and trained end-to-end along with the network (using the same initialization type used for the latent array's position encoding, see Appendix Sec. C). Because the position encodings used here are unaware of the structure of the input, it makes no difference whether inputs are permuted before or after the position encoding is constructed. We found that the network with 8 cross-attends had stability issues when learned position encodings are used, so we report results from a network with a single cross-attend.

On the face of it, this experiment may appear contrived – we know the grid structure, so why don't we use it? But the permuted settings provides a convenient model of the challenges presented by modalities that are challenging and large-scale (like ImageNet) but aren't naturally mapped to a 2D grid (e.g. point clouds, olfactory inputs, touch inputs, Lidar, etc.) or that include modalities that don't share a common grid (e.g. images + language, video + audio, somatosensory inputs + motor feedback, etc.).

Attention maps. Fig. 3 visualizes the attention maps at several cross-attention modules for a sample image from ImageNet (we include additional results in the Appendix). Each attention map shows the output of the QK^T operation at each of the model's 512 latent indices and each input pixel. We visualize attention maps before the softmax, as the

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).

softmax outputs are very sparse and hard to interpret. This model uses unshared weights in its initial cross-attention, but shares weights for all subsequent layers. The initial and later cross-attention layers produce qualitatively different attention maps: while the early modules shows clear traces of the input image (the dog pops out in many attention maps), the attention maps of later modules manifest as high-frequency plaid lattices. While the attention maps for modules 2 and 7 show similar structure, the specific details of corresponding maps do vary, which suggests the network attends to different sets of pixels at subsequent stages. The banded, variable-frequency structure of the attention maps appears to reflect the spatial frequency structure of the Fourier feature position encodings used on ImageNet. This tartan-like pattern is not present in networks with fully learned position encodings, suggesting it is at least in part due to the Fourier features.

4.2. Audio and video – AudioSet

We experimented with audio event classification in video using AudioSet (Gemmeke et al., 2017), a large dataset with 1.7M 10s long training videos and 527 classes. Videos may have multiple labels so we use a sigmoid cross entropy loss and evaluate using mean average precision (mAP). We evaluate the Perceiver using audio (using either the raw audio waveform or mel spectrogram), video, and audio + video as inputs. We sample 32-frame clips (1.28s at 25fps) in training; for evaluation we split the videos into 16 overlapping 32-frame clips, covering the whole 10s, and average the scores. We train models for 100 epochs.

Given the scale of the dataset we used a faster version of the ImageNet model with only 2 attention iterations instead of 8, but 8 self-attention layers per Transformer block instead of 6. We omit weight sharing to compensate for the smaller size. We experimented briefly with temporal unrolling – e.g. processing one frame per cross-attend – and found that it worked well and efficiently for video, but hurt performance for audio. Audio may require longer attention context.

Audio only. We use audio sampled at 48Khz resulting in 61,440 audio samples over 1.28s of video. We experimented with two settings: in the first we divide the raw signal into segments of 128 elements, for a total of 480 128-d vectors and input these to the Perceiver; the second setting uses a mel spectrogram resulting in 4800 inputs to the Perceiver, once flattened. As augmentations, for raw audio we simply sample in time, consistently with the video sampling. For spectrograms we use also specaugment (Park et al., 2019).

Video. A full 32 frame clip at 224x224 resolution has more than 2 million pixels. We experimented using tiny space-time patches with dimensions 2x8x8, resulting in a total of 12,544 inputs to the Perceiver. We compute Fourier features for horizontal, vertical and time coordinates (scaled to [-1, 1]), and concatenated them with the RGB values. We use the same model as in the audio experiments but now taking space-time patches as input rather than audio. We performed color augmentation, inception-type resizing, randomly flipping, and cropped to 224x224 resolution.

Audio + video. In this experiment we feed the Perceiver both the 12,544 space-time patches and either 480 raw audio vectors or 4,800 spectrogram values. Since modalities are fused at input, audio and video inputs need to have the same number of channels. We achieve this by concatenating a learned, modality-specific encoding to each input. As video has more channels, we use an embedding of size 4 for video inputs and make the audio encoding as large as necessary for the input channels between the two input arrays. This encoding doubles as a modality-specific position encoding (as discussed in Sec. 3.2), and we found it worked better than simply passing the audio encoding through a linear layer to match the video. Another thing that proved useful was **video dropout** – entirely zeroing out the video stream during training with some probability – a 30% probability for each example in each batch worked well. This may help the network to not overfit to video: these inputs provide a larger but less discriminative signal on Audioset. We observed a more than 3% improvement by using this proce-

THIS IS WHAT THEY "WOULD LIKE" TO AVOID RIGHT?

NO REASON WHY IT IS BENEFICIAL

dure; without it the spectrogram-based model scored 39.9% mAP (vs. 43.2%) and the raw audio model scored 39.7% (vs. 43.5%). After the ICML camera-ready deadline we tuned the spectrogram model further, and improved results to 44.2 by turning off specaugment and also dropping the spectrogram modality with 10% probability.

Results. Table 3 shows that the Perceiver obtains near state-of-the-art results on both video- and audio-only experiments. On raw audio the Perceiver gets 38.4, better than most ConvNet models except CNN-14 (Kong et al., 2020) which uses extra AugMix (Hendrycks et al., 2019) and class-balances the data – we hope to incorporate this in future work. Without these improvements the CNN-14 model does slightly worse than the Perceiver (37.5 mAP). Most previous methods use spectrograms as input but we find we can obtain similar performance even when using raw audio.

Audio+video fusion leads to solid improvements over single modalities (and outperforms specialized fusion optimization approaches (Wang et al., 2020c)) but is still lower than the state-of-the-art approach that uses separate models with late fusion (Fayek & Kumar, 2020). We will investigate this in future work. We visualize video and audio attention maps in Appendix Sec. E.

4.3. Point clouds – ModelNet40

ModelNet40 (Wu et al., 2015) is a dataset of point clouds derived from 3D triangular meshes spanning 40 object categories. The task is to predict the class of each object, given the coordinates of $\sim$ 2000 points in 3D space. ModelNet is small compared to other datasets used in our experiments: it has 9,843 training examples and 2,468 testing examples. To apply our model, we first preprocess point clouds by zero-centering them. To augment in training we apply random per-point scaling (between 0.99 and 1.01) followed by zero-mean and unit-cube normalization. We also explored random per-point translation (between -0.02 and 0.02) and random point-cloud rotation, but we found this did not improve performance.

We used an architecture with 2 cross-attentions and 6 self-attention layers for each block and otherwise used the same architectural settings as ImageNet. We used a higher maximum frequency than for image data to account for the irregular sampling structure of point clouds - we used a max frequency of 1120 (10$\times$ the value used on ImageNet). We obtained the best results using 64 frequency bands, and we noticed that values higher than 256 generally led to more severe overfitting. We used a batch size of 512 and trained with LAMB with a constant learning rate of 1×10^{-3}: models saturated in performance within 50,000 training steps.

Note that state-of-the-art methods on this benchmark are quite small and specialized and typically perform much more sophisticated data augmentation / feature engineering procedures, including fitting surfaces to the point clouds and using face normals as additional points (Qi et al., 2017). Here we are mostly interested in comparing to more generic models such as the ImageNet baselines and to assess how the various models deal with data that does not conform to a grid. Results of the Perceiver compared to the baselines are shown in Tab. 4. We arrange each point cloud into a 2D grid randomly, then feed it through each model. For ViT we varied the size of the patch size used at input.

	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.

5. Discussion

We have presented the Perceiver, a Transformer-based model that scales to more than a hundred thousand inputs. This opens new avenues for general perception architectures that make few assumptions about their inputs and that can handle arbitrary sensor configurations, while enabling fusion of information at all levels.

With great flexibility comes great overfitting, and many of our design decisions were made to mitigate this. In future work, we would like to pre-train our image classification model on very large scale data (Dosovitskiy et al., 2021). We obtain strong results on the large AudioSet dataset, which has 1.7M examples and where the Perceiver performed competitively with strong and recent state-of-the-art entries on audio, video and both combined. On ImageNet the model performs on par with ResNet-50 and ViT. When comparing these models across all different modalities and combinations considered in the paper, the Perceiver does best overall.

While we reduced the amount of modality-specific prior knowledge in the model, we still employ modality-specific augmentation and position encoding. End-to-end modality-agnostic learning remains an interesting research direction.

Acknowledgements

We are grateful to Sander Dieleman and Matt Botvinick for reviewing drafts of the paper, to Adrià Recasens Continente and Luyu Wang for help with AudioSet (especially Luyu for identifying an evaluation bug we had), and to Chris Burgess, Fede Carnevale, Mateusz Malinowski, Loïc Matthey, David Pfau, Adam Santoro, Evan Shelhamer, Greg Wayne, Chen Yan, Daniel Zoran and others at DeepMind for helpful conversations and suggestions. We thank Irwan Bello, James Betker, Andreas Kirsch, Christian Szegedy, Weidi Xie and others for comments on an earlier draft.

References

Adelson, E. H. and Bergen, J. R. Spatiotemporal energy models for the perception of motion. *Journal of the Optical Society of America A*, 2(2):284–299, 1985.

Alayrac, J.-B., Recasens, A., Schneider, R., Arandjelović, R., Ramapuram, J., De Fauw, J., Smaira, L., Dieleman, S., and Zisserman, A. Self-supervised multimodal versatile networks. In *Proceedings of Neural Information Processing Systems (NeurIPS)*, 2020.

Arandjelovic, R. and Zisserman, A. Objects that sound. In *Proceedings of the European conference on computer vision (ECCV)*, 2018.

Ba, J. L., Kiros, J. R., and Hinton, G. E. Layer normalization. *arXiv preprint arXiv:1607.06450*, 2016.

Babuschkin, I., Baumli, K., Bell, A., Bhupatiraju, S., Bruce, J., Buchlovsky, P., Budden, D., Cai, T., Clark, A., Danihelka, I., Fantacci, C., Godwin, J., Jones, C., Hennigan, T., Hessel, M., Kapturowski, S., Keck, T., Kemaev, I., King, M., Martens, L., Mikulik, V., Norman, T., Quan, J., Papamakarios, G., Ring, R., Ruiz, F., Sanchez, A., Schneider, R., Sezener, E., Spencer, S., Srinivasan, S., Stokowiec, W., and Viola, F. The DeepMind JAX Ecosystem, 2020. URL http://github.com/deepmind.

Bahdanau, D., Cho, K., and Bengio, Y. Neural machine translation by jointly learning to align and translate. In *Proceedings of International Conference on Learning Representations (ICLR)*, 2015.

Bello, I., Zoph, B., Vaswani, A., Shlens, J., and Le, Q. V. Attention augmented convolutional networks. In *Proceedings of IEEE International Conference on Computer Vision (ICCV)*, 2019.

Beltagy, I., Peters, M. E., and Cohan, A. Longformer: The long-document Transformer. *arXiv preprint arXiv:2004.05150*, 2020.

Borenstein, E., Sharon, E., and Ullman, S. Combining top-down and bottom-up segmentation. In *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, 2004.

Bradbury, J., Frostig, R., Hawkins, P., Johnson, M. J., Leary, C., Maclaurin, D., Necula, G., Paszke, A., VanderPlas, J., Wanderman-Milne, S., and Zhang, Q. JAX: composable transformations of Python+NumPy programs, 2018. URL http://github.com/google/jax.

Brock, A., De, S., Smith, S. L., and Simonyan, K. High-performance large-scale image recognition without normalization. *arXiv preprint arXiv:2102.06171*, 2021.

Bronstein, M., Bruna, J., LeCun, Y., Szlam, A., and Vandergheynst, P. Geometric deep learning: Going beyond Euclidean data. *IEEE Signal Processing Magazine*, 34 (4):18–42, 2017.

Carion, N., Massa, F., Synnaeve, G., Usunier, N., Kirillov, A., and Zagoruyko, S. End-to-end object detection with Transformers. In *Proceedings of European Conference on Computer Vision (ECCV)*, 2020.

Carreira, J., Agrawal, P., Fragkiadaki, K., and Malik, J. Human pose estimation with iterative error feedback. In *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016.

Chen, M., Radford, A., Child, R., Wu, J., Jun, H., Luan, D., and Sutskever, I. Generative pretraining from pixels. In *Proceedings of International Conference on Machine Learning (ICML)*, 2020a.

Chen, Y.-C., Li, L., Yu, L., El Kholy, A., Ahmed, F., Gan, Z., Cheng, Y., and Liu, J. UNITER: Learning UNiversal Image-TExt Representations. In *Proceedings of European Conference on Computer Vision (ECCV)*, 2020b.

Child, R., Gray, S., Radford, A., and Sutskever, I. Generating long sequences with sparse Transformers. *arXiv preprint arXiv:1904.10509*, 2019.

Choromanski, K., Likhosherstov, V., Dohan, D., Song, X., Gane, A., Sarlos, T., Hawkins, P., Davis, J., Mohiuddin, A., Kaiser, L., Belanger, D., Colwell, L., and Weller, A. Rethinking attention with Performers. In *Proceedings of International Conference on Learning Representations (ICLR)*, 2021.

Cireşan, D. C., Meier, U., Masci, J., Gambardella, L. M., and Schmidhuber, J. High-performance neural networks for visual object classification. Technical report, IDSIA, 2011.

Cohen, T. and Welling, M. Group equivariant convolutional networks. In *Proceedings of International Conference on Machine Learning (ICML)*, 2016.

Cordonnier, J.-B., Loukas, A., and Jaggi, M. On the relationship between self-attention and convolutional layers. In *Proceedings of International Conference on Learning Representations (ICLR)*, 2020.

Correia, G. M., Niculae, V., and Martins, A. F. Adaptively sparse Transformers. In *Conference on Empirical Methods in Natural Language Processing (EMNLP)*, 2019.

Cubuk, E. D., Zoph, B., Shlens, J., and Le, Q. V. Randaugment: Practical automated data augmentation with a reduced search space. In *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, 2020.

Dai, Z., Yang, Z., Yang, Y., Carbonell, J., Le, Q. V., and Salakhutdinov, R. Transformer-XL: Attentive language models beyond a fixed-length context. In *Annual Meetings of the Association for Computational Linguistics*, 2019.

Dehghani, M., Gouws, S., Vinyals, O., Uszkoreit, J., and Kaiser, L. Universal Transformers. In *Proceedings of International Conference on Learning Representations (ICLR)*, 2019.

Deng, J., Dong, W., Socher, R., Li, L.-J., Li, K., and Fei-Fei, L. ImageNet: A large-scale hierarchical image database. In *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2009.

Devlin, J., Chang, M.-W., Lee, K., and Toutanova, K. BERT: Pre-training of deep bidirectional Transformers for language understanding. In *Annual Conference of the North American Chapter of the Association for Computational Linguistics (NAACL)*, 2019.

Dosovitskiy, A., Beyer, L., Kolesnikov, A., Weissenborn, D., Zhai, X., Unterthiner, T., Dehghani, M., Minderer, M., Heigold, G., Gelly, S., et al. An image is worth 16x16 words: Transformers for image recognition at scale. In *Proceedings of International Conference on Learning Representations (ICLR)*, 2021.

Esteves, C., Allen-Blanchette, C., Makadia, A., and Daniilidis, K. Learning SO(3) equivariant representations with spherical CNNs. In *Proceedings of European Conference on Computer Vision (ECCV)*, 2018.

Fayek, H. M. and Kumar, A. Large scale audiovisual learning of sounds with weakly labeled data. In *Proceedings of International Joint Conference on Artificial Intelligence*, 2020.

Felleman, D. J. and Essen, D. C. V. Distributed hierarchical processing in the primate cerebral cortex. *Cerebral Cortex*, 1(1):1–47, 1991.

Ford, L., Tang, H., Grondin, F., and Glass, J. R. A deep residual network for large-scale acoustic scene analysis. In *Proceedings of Interspeech*, pp. 2568–2572, 2019.

Fukushima, K. Neocognitron: A self-organizing neural network model for a mechanism of pattern recognition unaffected by shift in position. *Biological Cybernetics*, (36):193—-202, 1980.

Gehring, J., Auli, M., Grangier, D., Yarats, D., and Dauphin, Y. N. Convolutional sequence to sequence learning. In *Proceedings of International Conference on Machine Learning (ICML)*, 2017.

Gemmeke, J. F., Ellis, D. P., Freedman, D., Jansen, A., Lawrence, W., Moore, R. C., Plakal, M., and Ritter, M. Audio Set: An ontology and human-labeled dataset for audio events. In *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2017.

Girdhar, R., Carreira, J., Doersch, C., and Zisserman, A. Video action Transformer network. In *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019.

Goyal, A., Didolkar, A., Lamb, A., Badola, K., Ke, N. R., Rahaman, N., Binas, J., Blundell, C., Mozer, M., and Bengio, Y. Coordination among neural modules through a shared global workspace. *arXiv preprint arXiv:2103.01197*, 2021.

Graves, A., Mohamed, A., and Hinton, G. Speech recognition with deep recurrent neural networks. In *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2013.

Graves, A., Wayne, G., and Danihelka, I. Neural Turing machines. *arXiv preprint arXiv:1410.5401*, 2014.

Guo, M.-H., Cai, J.-X., Liu, Z.-N., Mu, T.-J., Martin, R. R., and Hu, S.-M. Pct: Point Cloud Transformer. *arXiv preprint arXiv:2012.09688*, 2020.

He, K., Zhang, X., Ren, S., and Sun, J. Deep residual learning for image recognition. In *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016.

Hendrycks, D. and Gimpel, K. Gaussian error linear units (GELUs). *arXiv preprint arXiv:1606.08415*, 2016.

Hendrycks, D., Mu, N., Cubuk, E. D., Zoph, B., Gilmer, J., and Lakshminarayanan, B. AugMix: A simple data processing method to improve robustness and uncertainty. *arXiv preprint arXiv:1912.02781*, 2019.

Ho, J., Kalchbrenner, N., Weissenborn, D., and Salimans, T. Axial attention in multidimensional Transformers. *arXiv preprint arXiv:1912.12180*, 2019.

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

Hu, P. and Ramanan, D. Bottom-up and top-down reasoning with hierarchical rectified gaussians. In *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016.

Kaiser, L., Gomez, A. N., Shazeer, N., Vaswani, A., Parmar, N., Jones, L., and Uszkoreit, J. One model to learn them all. *arXiv preprint arXiv:1706.05137*, 2017.

Kandel, E., Schwartz, J., Jessell, T., Siegelbaum, S., and Hudspeth, A. *Principles of Neural Science*. McGraw-Hill Education, Fifth edition, 2012.

Kant, I. *Critique of Pure Reason*. 1781.

Kaplan, J., McCandlish, S., Henighan, T., Brown, T. B., Chess, B., Child, R., Gray, S., Radford, A., Wu, J., and Amodei, D. Scaling laws for neural language models. *arXiv preprint arXiv:2001.08361*, 2020.

Karpathy, A., Toderici, G., Shetty, S., Leung, T., Sukthankar, R., and Fei-Fei, L. Large-scale video classification with convolutional neural networks. In *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2014.

Katharopoulos, A., Vyas, A., Pappas, N., and Fleuret, F. Transformers are RNNs: Fast autoregressive Transformers with linear attention. In *Proceedings of International Conference on Machine Learning (ICML)*, 2020.

Kitaev, N., Kaiser, L., and Levskaya, A. Reformer: The efficient Transformer. In *Proceedings of International Conference on Learning Representations (ICLR)*, 2020.

Köhler, W. Gestalt psychology. *Psychologische Forschung*, 31(1):XVIII–XXX, 1967.

Kong, Q., Xu, Y., Wang, W., and Plumbley, M. D. Audio set classification with attention model: A probabilistic perspective. In *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2018.

Kong, Q., Cao, Y., Iqbal, T., Wang, Y., Wang, W., and Plumbley, M. D. Panns: Large-scale pretrained audio neural networks for audio pattern recognition. *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, 2020.

Krizhevsky, A., Sutskever, I., and Hinton, G. E. Imagenet classification with deep convolutional neural networks. *Advances in neural information processing systems*, 25, 2012.

Kumar, M. P., Ton, P., and Zisserman, A. Obj cut. In *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2005.

Lan, Z., Chen, M., Goodman, S., Gimpel, K., Sharma, P., and Soricut, R. Albert: A lite BERT for self-supervised learning of language representations. In *Proceedings of International Conference on Learning Representations (ICLR)*, 2020.

LeCun, Y., Bottou, L., Bengio, Y., and Haffner, P. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11):2278–2324, 1998.

LeCun, Y., Bengio, Y., and Hinton, G. Deep learning. *Nature*, 521(7553):436–444, 2015.

Lee, J., Lee, Y., Kim, J., Kosiorek, A., Choi, S., and Teh, Y. W. Set Transformer: A framework for attention-based permutation-invariant neural networks. In *Proceedings of International Conference on Machine Learning (ICML)*, 2019.

Lee, M. A., Zhu, Y., Zachares, P., Tan, M., Srinivasan, K., Savarese, S., Fei-Fei, L., Garg, A., and Bohg, J. Making sense of vision and touch: Learning multimodal representations for contact-rich tasks. *IEEE Transactions on Robotics*, 36(3), 2020.

Lin, X., Ma, L., Liu, W., and Chang, S.-F. Context-gated convolution. In *Proceedings of European Conference on Computer Vision (ECCV)*, 2020.

Locatello, F., Weissenborn, D., Unterthiner, T., Mahendran, A., Heigold, G., Uszkoreit, J., Dosovitskiy, A., and Kipf, T. Object-centric learning with slot attention. In *Proceedings of Neural Information Processing Systems (NeurIPS)*, 2020.

Lollo, V. D., Enns, J. T., and Rensink, R. A. Competition for consciousness among visual events: The psychophysics of reentrant visual processes. *Journal of Experimental Psychology*, 129(4):481–507, 2000.

Mildenhall, B., Srinivasan, P. P., Tancik, M., Barron, J. T., Ramamoorth, R., and Ng, R. Nerf: Representing scenes as neural radiance fields for view synthesis. In *Proceedings of European Conference on Computer Vision (ECCV)*, 2020.

Nyquist, H. Certain topics in telegraph transmission theory. *Transactions of the American Institute of Electrical Engineers*, 47(2):617–644, 1928.

Olshausen, B. A., Anderson, C. H., and Van Essen, D. C. A neurobiological model of visual attention and invariant pattern recognition based on dynamic routing of information. *Journal of Neuroscience*, 13(11):4700–4719, 1993.

Park, D. S., Chan, W., Zhang, Y., Chiu, C.-C., Zoph, B., Cubuk, E. D., and Le, Q. V. SpecAugment: A simple data augmentation method for automatic speech recognition. In *Proceedings of Interspeech*, 2019.

Parmar, N., Vaswani, A., Uszkoreit, J., Kaiser, L., Shazeer, N., Ku, A., and Tran, D. Image Transformer. In *Proceedings of International Conference on Machine Learning (ICML)*, 2018.

Peng, H., Pappas, N., Yogatama, D., Schwartz, R., Smith, N., and Kong, L. Random feature attention. In *Proceedings of International Conference on Learning Representations (ICLR)*, 2021.

Qi, C. R., Yi, L., Su, H., and Guibas, L. J. PointNet++: Deep hierarchical feature learning on point sets in a metric space. In *Proceedings of Neural Information Processing Systems (NeurIPS)*, 2017.

Radford, A., Wu, J., Child, R., Luan, D., Amodei, D., and Sutskever, I. Language models are unsupervised multitask learners. Technical report, OpenAI, 2019.

Rae, J. W., Potapenko, A., Jayakumar, S. M., Hillier, C., and Lillicrap, T. P. Compressive transformers for long-range sequence modelling. In *Proceedings of International Conference on Learning Representations (ICLR)*, 2020.

Ramachandran, P., Parmar, N., Vaswani, A., Bello, I., Levskaya, A., and Shlens, J. Stand-alone self-attention in vision models. In *Proceedings of Neural Information Processing Systems (NeurIPS)*, 2019.

Roy, A., Saffar, M., Vaswani, A., and Grangier, D. Efficient content-based sparse attention with routing Transformers. In *Transactions of the Association for Computational Linguistics (TACL)*, 2020.

Santoro, A., Faulkner, R., Raposo, D., Rae, J., Chrzanowski, M., Weber, T., Wierstra, D., Vinyals, O., Pascanu, R., and Lillicrap, T. Relational recurrent neural networks. In *Proceedings of Neural Information Processing Systems (NeurIPS)*, 2018.

Shi, J. and Malik, J. Normalized cuts and image segmentation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 22(8):888–905, 2000.

Simoncelli, E. P. and Heeger, D. J. A model of neuronal responses in visual area MT. *Vision Research*, 38(5): 743–761, 1998.

Simonyan, K. and Zisserman, A. Very deep convolutional networks for large-scale image recognition. In *Proceedings of International Conference on Learning Representations (ICLR)*, 2015.

Srinivas, A., Lin, T.-Y., Parmar, N., Shlens, J., Abbeel, P., and Vaswani, A. Bottleneck Transformers for visual recognition. *arXiv preprint arXiv:2101.11605*, 2021.

Stanley, K. O. Compositional pattern producing networks: A novel abstraction of development. *Genetic programming and evolvable machines*, 8(20):131 – 162, 2007.

Sukhbaatar, S., Grave, E., Bojanowski, P., and Joulin, A. Adaptive attention span in Transformers. In *Annual Meetings of the Association for Computational Linguistics*, 2019.

Sutton, R. The bitter lesson. *http://www. incompleteideas. net/IncIdeas/BitterLesson*, 2019.

Szegedy, C., Liu, W., Jia, Y., Sermanet, P., Reed, S., Anguelov, D., Erhan, D., Vanhoucke, V., and Rabinovich, A. Going deeper with convolutions. In *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2015.

Tancik, M., Srinivasan, P. P., Mildenhall, B., Fridovich-Keil, S., Raghavan, N., Singhal, U., Ramamoorthi, R., Barron, J. T., and Ng, R. Fourier features let networks learn high frequency functions in low dimensional domains. In *Proceedings of Neural Information Processing Systems (NeurIPS)*, 2020.

Tay, Y., Dehghani, M., Bahri, D., and Metzler, D. Efficient Transformers: A survey. *arXiv preprint arXiv:2009.06732*, 2020.

Tay, Y., Bahri, D., Metzler, D., Juan, D.-C., Zhao, Z., and Zheng, C. Synthesizer: Rethinking self-attention in Transformer models. In *Proceedings of International Conference on Machine Learning (ICML)*, 2021a.

Tay, Y., Dehghani, M., Abnar, S., Shen, Y., Bahri, D., Pham, P., Rao, J., Yang, L., Ruder, S., and Metzler, D. Long Range Arena: A benchmark for efficient Transformers. In *Proceedings of International Conference on Learning Representations (ICLR)*, 2021b.

Touvron, H., Cord, M., Douze, M., Massa, F., Sablayrolles, A., and Jégou, H. Training data-efficient image Transformers & distillation through attention. *arXiv preprint arXiv:2012.12877*, 2020.

Tran, D., Bourdev, L., Fergus, R., Torresani, L., and Paluri, M. Learning spatiotemporal features with 3D convolutional networks. In *Proceedings of IEEE International Conference on Computer Vision (ICCV)*, 2015.

Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, L., and Polosukhin, I. Attention is all you need. In *Proceedings of Neural Information Processing Systems (NeurIPS)*, 2017.

Wang, H., Zhu, Y., Green, B., Adam, H., Yuille, A., and Chen, L.-C. Axial-DeepLab: Stand-alone axial-attention for panoptic segmentation. In *Proceedings of European Conference on Computer Vision (ECCV)*, 2020a.

Wang, H., Zhu, Y., Adam, H., Yuille, A., and Chen, L.-C. Max-deeplab: End-to-end panoptic segmentation with mask transformers. In *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2021.

Wang, S., Li, B. Z., Khabsa, M., Fang, H., and Ma, H. Linformer: Self-attention with linear complexity. *arXiv preprint arXiv:2006.04768*, 2020b.

Wang, W., Tran, D., and Feiszli, M. What makes training multi-modal classification networks hard? In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 12695–12705, 2020c.

Wang, X., Girshick, R., Gupta, A., and He, K. Non-local neural networks. In *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2018.

Weston, J., Chopra, S., and Bordes, A. Memory networks. In *Proceedings of International Conference on Learning Representations (ICLR)*, 2015.

Wolfe, J. M., Reinecke, A., and Brawn, P. Why don't we see changes? the role of attentional bottlenecks and limited visual memory. *Visual Cognition*, 14(4–8):749–780, 2006.

Wu, B., Xu, C., Dai, X., Wan, A., Zhang, P., Yan, Z., Tomizuka, M., Gonzalez, J., Keutzer, K., and Vajda, P. Visual Transformers: Token-based image representation and processing for computer vision. *arXiv preprint arXiv:2006.03677*, 2020.

Wu, Z., Song, S., Khosla, A., Yu, F., Zhang, L., Tang, X., and Xiao, J. 3D shapenets: A deep representation for volumetric shapes. In *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2015.

Xiao, F., Lee, Y. J., Grauman, K., Malik, J., and Feichtenhofer, C. Audiovisual slowfast networks for video recognition. *arXiv preprint arXiv:2001.08740*, 2020.

Xiong, Y., Zeng, Z., Chakraborty, R., Tan, M., Fung, G., Li, Y., and Singh, V. Nyströmformer: A Nyström-based algorithm for approximating self-attention. In *Proceedings of AAAI Conference on Artificial Intelligence*, 2021.

Yang, Y., Zhong, Z., Shen, T., and Lin, Z. Convolutional neural networks with alternately updated clique. In *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2018.

Ye, Z., Guo, Q., Gan, Q., Qiu, X., and Zhang, Z. BP-Transformer: Modelling long-range context via binary partitioning. *arXiv preprint arXiv:1911.04070*, 2019.

You, Y., Li, J., Reddi, S., Hseu, J., Kumar, S., Bhojanapalli, S., Song, X., Demmel, J., Keutzer, K., and Hsieh, C.-J. Large batch optimization for deep learning: Training bert in 76 minutes. In *Proceedings of International Conference on Learning Representations (ICLR)*, 2020.

Yu, C., Barsim, K. S., Kong, Q., and Yang, B. Multi-level attention model for weakly supervised audio classification. In *DCASE2018 Workshop on Detection and Classification of Acoustic Scenes and Events*, 2018.

Zhao, H., Jia, J., and Koltun, V. Exploring self-attention for image recognition. In *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2020.

Zoran, D., Chrzanowski, M., Huang, P.-S., Gowal, S., Mott, A., and Kohli, P. Towards robust image classification using sequential attention models. In *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2020.

Appendices

A. Extended related work

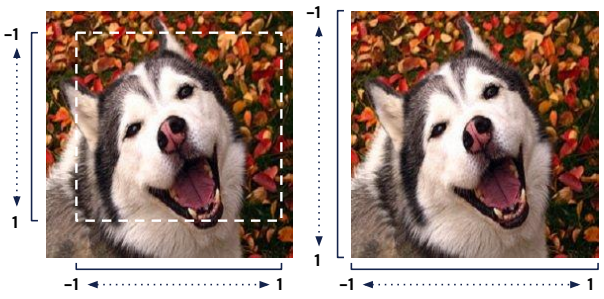
Efficient attention architectures, cont'd. Several strategies have been proposed to gain greater efficiency by modifying the internals of the Transformer's self-attention module, including using local or patchwise self-attention (Parmar et al., 2018; Ramachandran et al., 2019; Zhao et al., 2020; Sukhbaatar et al., 2019), using non-local, non-dense attention patterns (Ho et al., 2019; Wang et al., 2020a; Beltagy et al., 2020; Child et al., 2019; Correia et al., 2019; Ye et al., 2019; Roy et al., 2020), approximating or otherwise simplifying the matrix multiplications used in QKV attention (Choromanski et al., 2021; Peng et al., 2021; Kitaev et al., 2020; Xiong et al., 2021; Katharopoulos et al., 2020; Tay et al., 2021a), or by introducing bottlenecks into each module's computation (Lee et al., 2019; Wang et al., 2020b). The primary contribution of this body of work is a set of modules with similar flexibility to the Transformer's densely-connected self-attention block, but at sub-quadratic computational cost (see Tay et al. 2020; 2021b for more detailed reviews). The focus of our work is primarily on producing an architecture that is efficient as a whole and is suitable for many domains, rather than improving the complexity of the Transformer's self-attention module itself. In this sense, our work is complementary to this large and very interesting body of work, and it is likely that some of these approaches could be used to further increase the Perceiver's efficiency.

Relationship to the Set Transformer. The Set Transformer work (Lee et al., 2019) introduces two modules (called ISAB for "induced set attention block" and PMA for "pooling by multiheaded attention"), that function similarly to the cross-attention blocks we use here but are deployed in a different manner. ISAB is used to map an input array (interpreted as a set) to a low-dimensional array and immediately map it back to the input space. Stacking these blocks leads to an architecture that scales linearly in compute/memory with input size like the Perceiver's cross-attention module, but without the advantage of the Perceiver's latent array (which completely decouples the cost of the latent Transformer from the input size): a fully-ISAB model scales as $\mathcal{O}(LMN)$, rather than $\mathcal{O}(MN + LN^2)$, like the Perceiver (where M is the index dimension of the input, N the index dimension of the latent, and L the network depth).

PMA is used to map an input array to an output array with a sized determined by the task (e.g. 1 point for classification or 4 points for a 4-way clustering task). It is used to map to a target output size and not to induce a latent space. In contrast, the Perceiver's latent space has a size that is independent of the task (it is a hyperparameter, and typically much larger than the task output) and is designed specifically to facilitate the efficient construction of deep, latent Transformers. To use the Set Transformer terminology, a Perceiver directly feeds its input to a PMA-like block (or $\frac{1}{2}$ of an ISAB-like block) whose output size is relatively large (e.g. 512) and task-independent rather than determined by the task; it would be 1 (for classification) if used as proposed in the Set Transformer. This is followed by a very deep stack of (latent) self-attention blocks and a final average and project. In other words, Perceivers exploit similar primitives to the Set Transformer, but compose them differently, in service of building an architecture with improved scaling properties.

Cross-attention and attentional latents. More broadly, cross-attention has been used to augment Transformer architectures with attention to the longer-horizon past (Rae et al., 2020; Dai et al., 2019) and to produce architectures that write to and/or read from fixed-size arrays or memories (Santoro et al., 2018; Goyal et al., 2021), all while keeping the cost of each operation linear in the input size. We use cross-attention to induce a latent space for deep processing. This can be viewed as a fully attentional, domain-agnostic analogue of models that stacks self-attention on top of convolutional feature maps to perform cheap but global processing on top or in conjunction with otherwise spatially localized convolutional feature maps (e.g. Carion et al. 2020; Locatello et al. 2020; Wang et al. 2021).

Global, re-entrant processing. The Perceiver performs global computations from the first layer: although contemporary architectures typically first process locally, the notion of building perception systems using global processing has a long history (e.g. Köhler 1967; Shi & Malik 2000). When inputs grow very large, this may introduce a bandwidth bottleneck. By using multiple cross-attentions, the Perceiver can use a form of re-entrant processing to mitigate this effect, by allowing first-pass processing of an input to feed back and influence how the input is processed in subsequent passes. Re-entrant processing of this kind (sometimes referred to as top-down processing) has a long history in computer vision (Borenstein et al., 2004; Kumar et al., 2005; Carreira et al., 2016; Hu & Ramanan, 2016; Yang et al., 2018; Lin et al., 2020). There is widespread evidence that it plays an important role in human vision (e.g. Felleman & Essen 1991; Olshausen et al. 1993; Lollo et al. 2000), which is characterized by limited bandwidth input streams (Wolfe et al., 2006). In the Perceiver, attention to the full set of inputs can be influenced by a latent array produced by previous iterations of the model, allowing the model focus on subsets of inputs that are most promising in a soft way (Zoran et al., 2020).



(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.

B. Ablations

To illustrate the effect of various network hyperparameters, we considered a small Perceiver model and swept a number of options around it. Unlike ConvNets, each module in a Perceiver-based architecture takes as input the full input byte array. This makes it possible to sweep processing hyperparameters (e.g. depth, capacity, etc.), without reducing the effective receptive field size of the network as a whole. The base model did not share either self-attention or cross-attention parameters, used 8 heads per self-attention module, 4 self-attention modules per block, performed 2 cross-attends per image, and had a latent with an index dimension of 512 and a channel dimension 512. We used a small batch size of 64 across 32 TPUs to make sure all models fit comfortably in memory no matter how extreme the parameters. We trained all models for 5 million steps using a similar optimization procedure as in the main paper.

The results from a hyperparameter sweep centered on this base architecture are shown in Fig. 5. All results show top-1 accuracy on ImageNet. Consistent with our other experiments, these results suggest that increasing the size of the model tends to produce better results. The exception in this experiment was the number of latent dimensions, as the largest model showed signs of overfitting.

Similarly, we evaluated the effect of the latent array's initialization scale and the parameters of the Fourier frequency position encoding on ImageNet performance. The results of this experiment are shown in Fig. 6. These experiments use the full-sized ImageNet architecture, but were trained with a smaller batch size (256) and fewer TPUs (16) (for reasons of compute availability). These experiments suggest that standard and relatively small values for the initialization scale are best (values ≥ 1 may lead to instability), and generally suggest that a higher number of Fourier frequency bands and a higher maximum resolution (up to Nyquist) improve performance. We found that a scale of 1.0 worked best for initializing the position encoding: this value is used for the model reported in Tab. 2.

SELF-ATTENTION IS BENEFICIAL

# 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

CA SA INTERLEAVED

CA CA ... SA SA AT-START

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).

For all FLOPs numbers reported here, we report unfused multiply-adds

All FLOPS reported here give theoretical FLOPS with multiplies and accumulates counted as separate operations. This is the strategy used in (Kaplan et al., 2020) and elsewhere in the literature. Note that some other papers in the literature report FLOPS using fused multiply-accumulates: using this strategy will approximately cut our reported figures in half.

C. Architectural details

The Perceiver consists of two modules: a cross-attention module and a Transformer. In the cross-attention module, inputs are first processed with layer norm (Ba et al., 2016) before being passed through linear layers to produce each of the query, key, and value inputs to the QKV cross-attention operation. The queries, keys, and values have the same number of channels as the minimum of the input channels, which is typically the key/value input (i.e. 261 for ImageNet) The output of attention is passed through an additional linear layer to project it to the same number of channels in the query inputs (so it can be added residually).

The query inputs for the first cross-attention layer (e.g. the

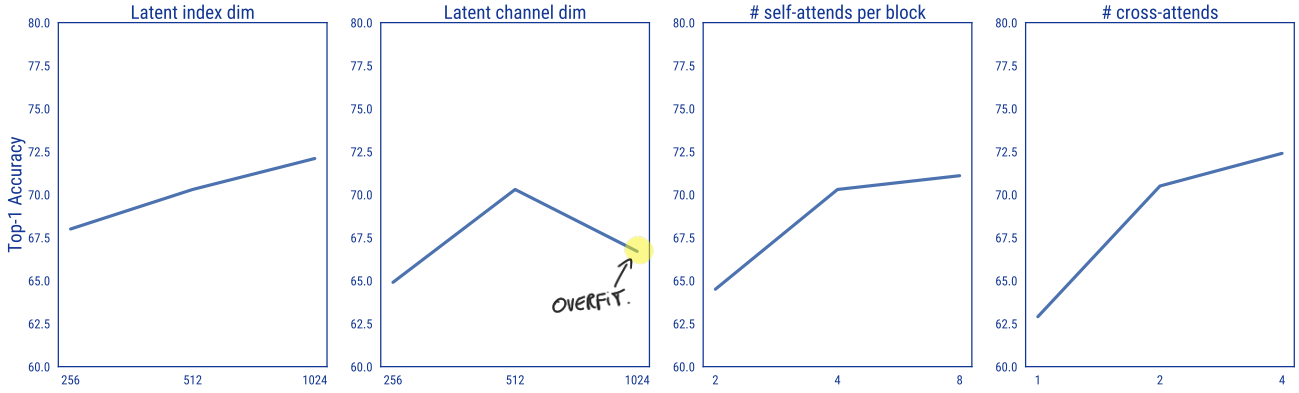


Figure 5. The effect of model hyperparameters, using a scaled-down Perceiver architecture on ImageNet. All plots show top-1 accuracy (higher is better). Increasing the size of the latent index dimension, the number of self-attends per block, and the number of cross-attends generally produced better results.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

left-most latent array in Fig. 1) are learned, per-element weights with the same shape as the latent array (e.g. for ImageNet, they are a 512×1024 array). These function like learned position encodings in the Transformer literature or like a learned initial state in the recurrent neural network (RNN) literature. The latent array is randomly initialized using a truncated normal distribution with mean 0, standard deviation 0.02, and truncation bounds [-2, 2]. Network performance is fairly robust to the scale of this initialization (see Fig. 6, left facet).

In the self-attention block, inputs are processed with layer norm and passed through query, key, and value layers before being used to compute QKV self-attention. The output is passed through another linear layer.

Each cross-attention and self-attention block is followed by a dense (multi-layer Perceptron) block. In the dense block, inputs are processed with layer norm, passed through a linear layer, activated with a GELU nonlinearity (Hendrycks & Gimpel, 2016), and passed through a final linear layer. We used dropout throughout the network in earlier experiments, but we found this led to degraded performance, so no dropout is used.

All linear layers (including query, key, and value layers and dense block layers) preserve the dimensionality of their inputs and are tiled over input index dimensions (i.e. applied as a 1×1 convolution).

To produce output logits, we average the output of the final latent self-attention module over the index dimension. This produces a single global summary vector, which we then project to the number of target classes using a single linear layer. This is the process used by e.g. ResNets to convert a convolutional feature map to logits (He et al., 2016).

As with other Transformer architectures, the Perceiver's Transformer has a fully residual design, and its input is always added to its output for further processing. This applies to cross-attention modules as well: the latent component of the input is added to its output. We give details on the hyperparameters used on different datasets in the main paper.

	Valid	Train	Params	FLOPs
No weight sharing	72.9	87.7	326.2M	707.2B
W/ weight sharing	**78.0**	79.5	44.9M	707.2B

Table 7. Weight sharing mitigates overfitting and leads to better validation performance on ImageNet. We show results (top-1 accuracy) for the best-performing ImageNet architecture (reported in Tables 1-2 of the main paper) on train and validation sets. This architecture uses 8 cross-attends and 6 blocks per latent Transformer. The model labeled "W/ weight sharing" shares weights between cross-attention modules 2-8 and between the corresponding blocks of all latent Transformers. The first cross-attention module uses its own, unshared weights.

D. Position encodings and Fourier features

Crop-relative coordinates. As described in the main paper, we found that generating position coordinates using cropped data rather than on the raw data was important to prevent excessive overfitting. We illustrate the cropping procedure on ImageNet in Fig. 4.

Fourier feature parameterizations. We choose the Fourier feature parameterization described in section 3.2 of the paper to allow us to intuitively set the maximum band when the sample rate of the input signal is regular and known. By setting the number of bands independently, we allow it be easily controlled in line with a computational budget: we generally found that more bands helped for a given architecture (assuming it fits in memory). For signals with irregular or very fine sampling, such as ModelNet40

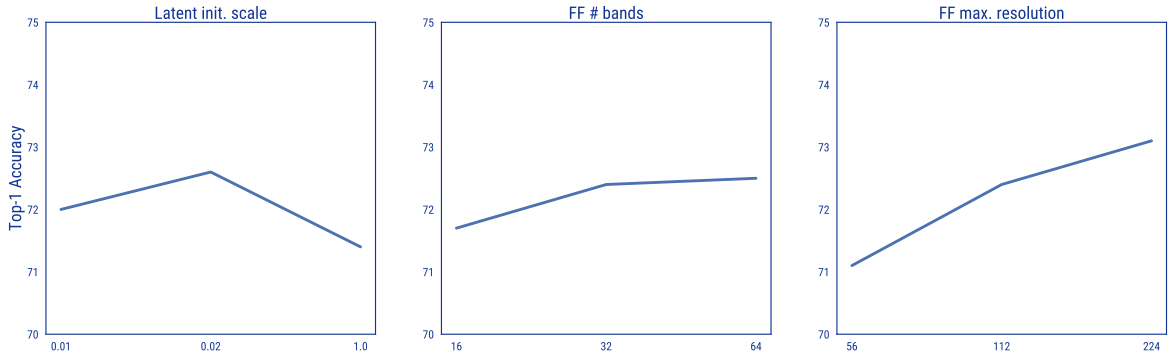


Figure 6. The effect of latent initialization scale and Fourier feature (FF) position encoding parameters on performance. All plots show top-1 accuracy (higher is better). The model with initialization scale of 0.1 diverged during training. Generally, increasing the number of bands and max resolution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

point clouds, the maximum band can also be treated as a hyperparameter. This is in contrast to the parameterization used in NeRF (Mildenhall et al., 2020), which produces very high frequencies if a moderate number of bands are used (e.g. the 64^{th} band would have a frequency of $2^{64} = 1.8e19$). Rather than tying the maximum frequency to the number of bands, our parameterization samples the spectrum more densely as more bands are added. Our parameterization is identical to the parameterization described in the original Transformer paper, except we express each band in terms of its frequency rather than its wavelength (we find this more natural in the context of signals like images) and we assume that input positions are in $[-1, 1]$ rather than $[0, s)$ for a sequence of length s.

E. Audiovisual attention maps

We visualize video and audio attention maps (respectively) for the first and second cross-attention module of a multimodal Perceiver model trained on AudioSet using 2x4x4 video patches and 61,440 raw audio samples

We visualize video attention maps similarly to static image attention maps (Fig. 3), but with the addition of a time dimension: each column shows the attention to the full image at a time step of the video. Because this AudioSet Perceiver takes space-time patches of shape time 2 $\times$ height 4 $\times$ width 4, the same attention is applied to pairs of subsequent frames. For visualization purposes, we show every other frame of the input video and attention maps: each attention map is applied to two video frames.

All attention maps of this network appear to be sensitive to both static and dynamic features of the input video. Some attention maps exhibit spatiotemporal structure reminiscent of the filters seen in spatiotemporal image processing (Adelson & Bergen, 1985; Simoncelli & Heeger, 1998). Because the Perceiver uses learned attention rather than a fixed bank of spatiotemporal filters, it can adapt its attention to the input content.

We visualize audio attention maps by displaying the mel-spectrogram and attention maps as images. Mel-spectrograms are plotted with time plotted on the x- and frequency on the y-axis. Although they are harder to interpret visually than the image attention maps, they appear to share a common structure of Fourier-frequency positional banding and content-related modulation.

F. Notes on changes from the original version

Our Audioset mAP results in the first arXiv version were flawed (and unfortunately higher) so we repeated and expanded those experiments and now provide correct numbers. The issue was that when computing AP using the sklearn package, we passed the matrix of class scores transposed to what the function expects – hence the number of classes and number of examples were switched.

We have slightly improved the results reported on ImageNet since the first version by (i) removing dropout, (ii) removing a linear layer that was originally (unintentionally) included following the initial latent array, and (iii) averaging before rather than after projecting when computing the output logits.

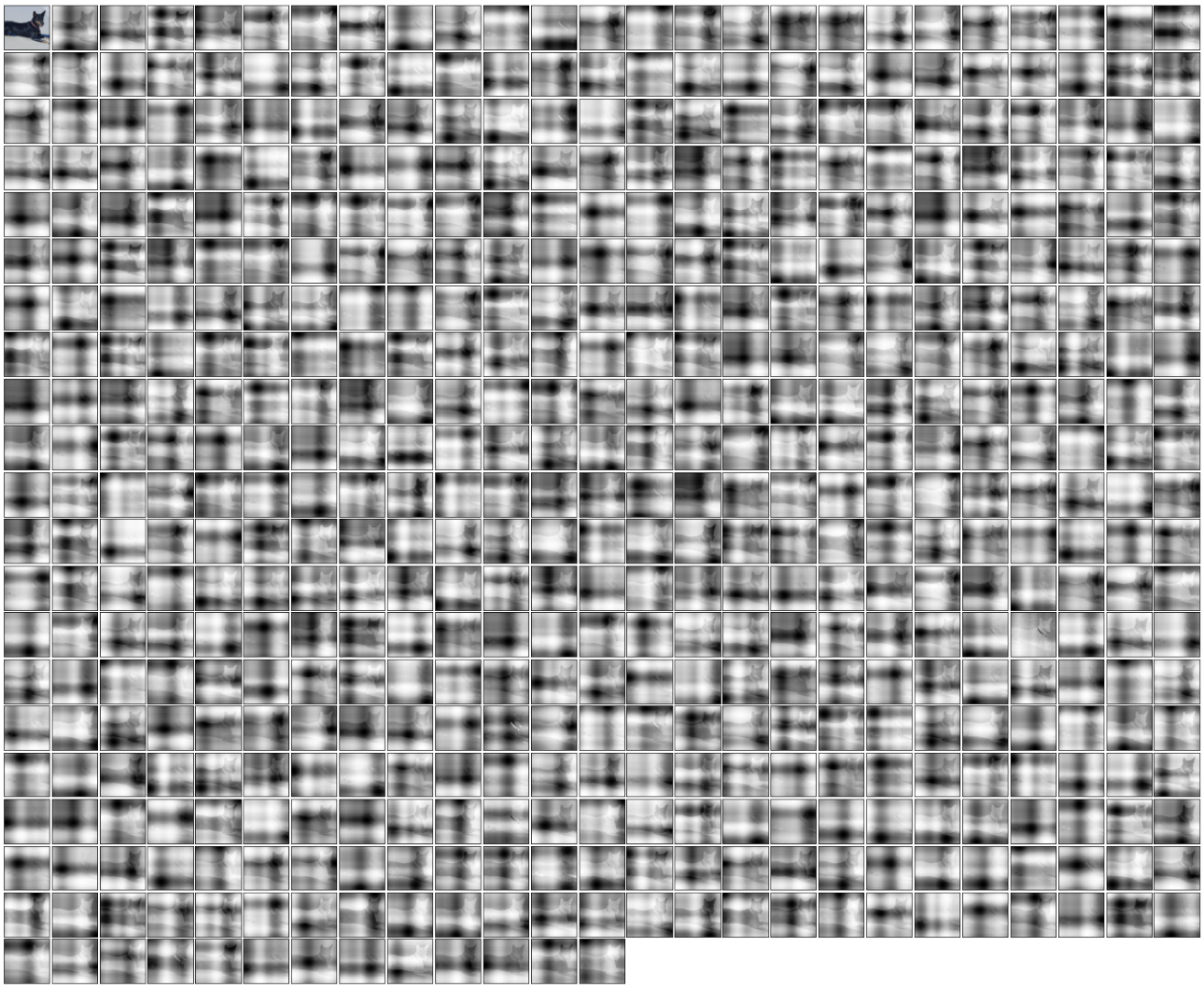


Figure 7. Example attention maps from the **first cross-attend** of an ImageNet network trained with **2D Fourier feature** position encodings.

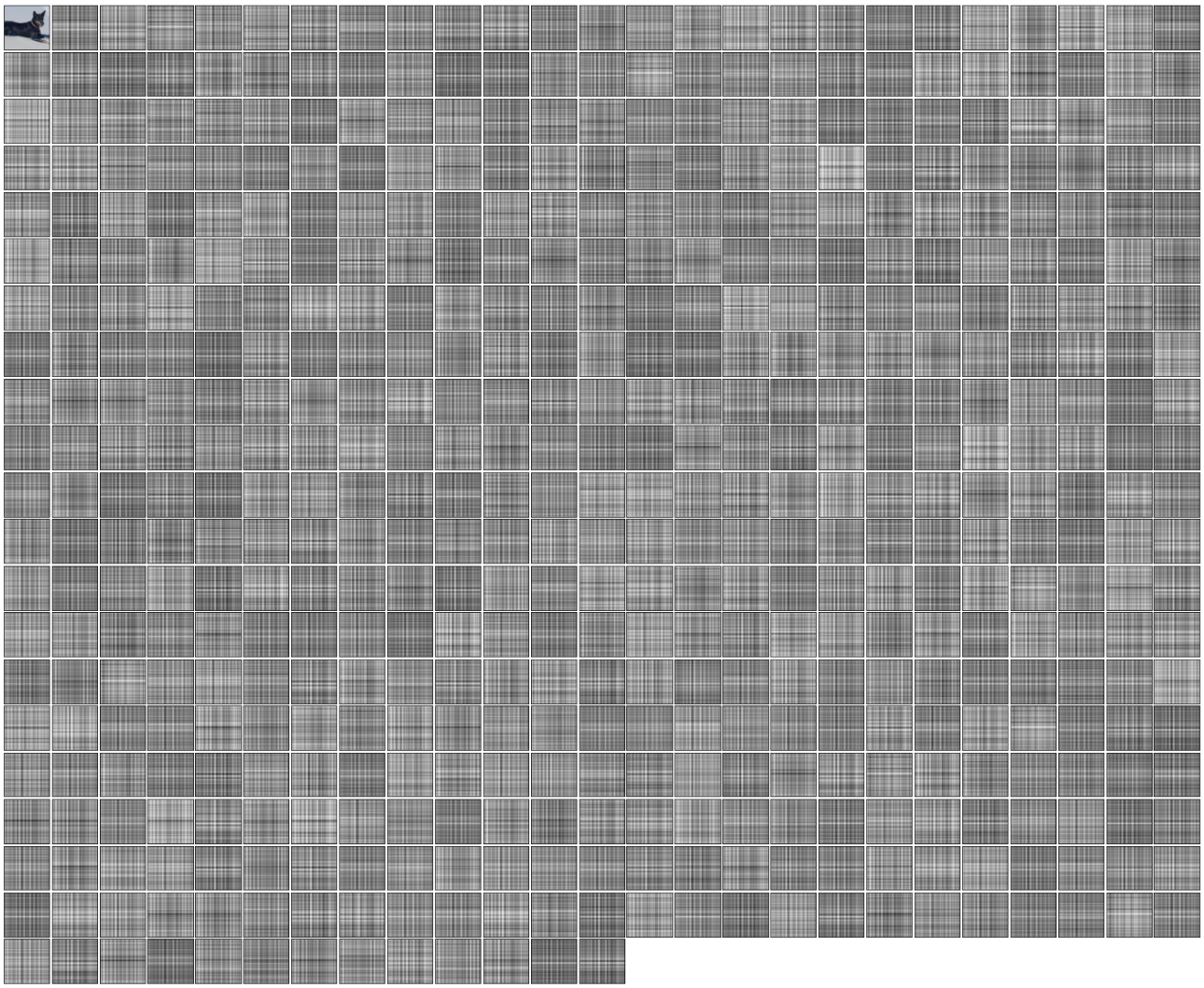


Figure 8. Example attention maps from the **eighth (final) cross-attend** of an ImageNet network trained with **2D Fourier feature** position encodings.

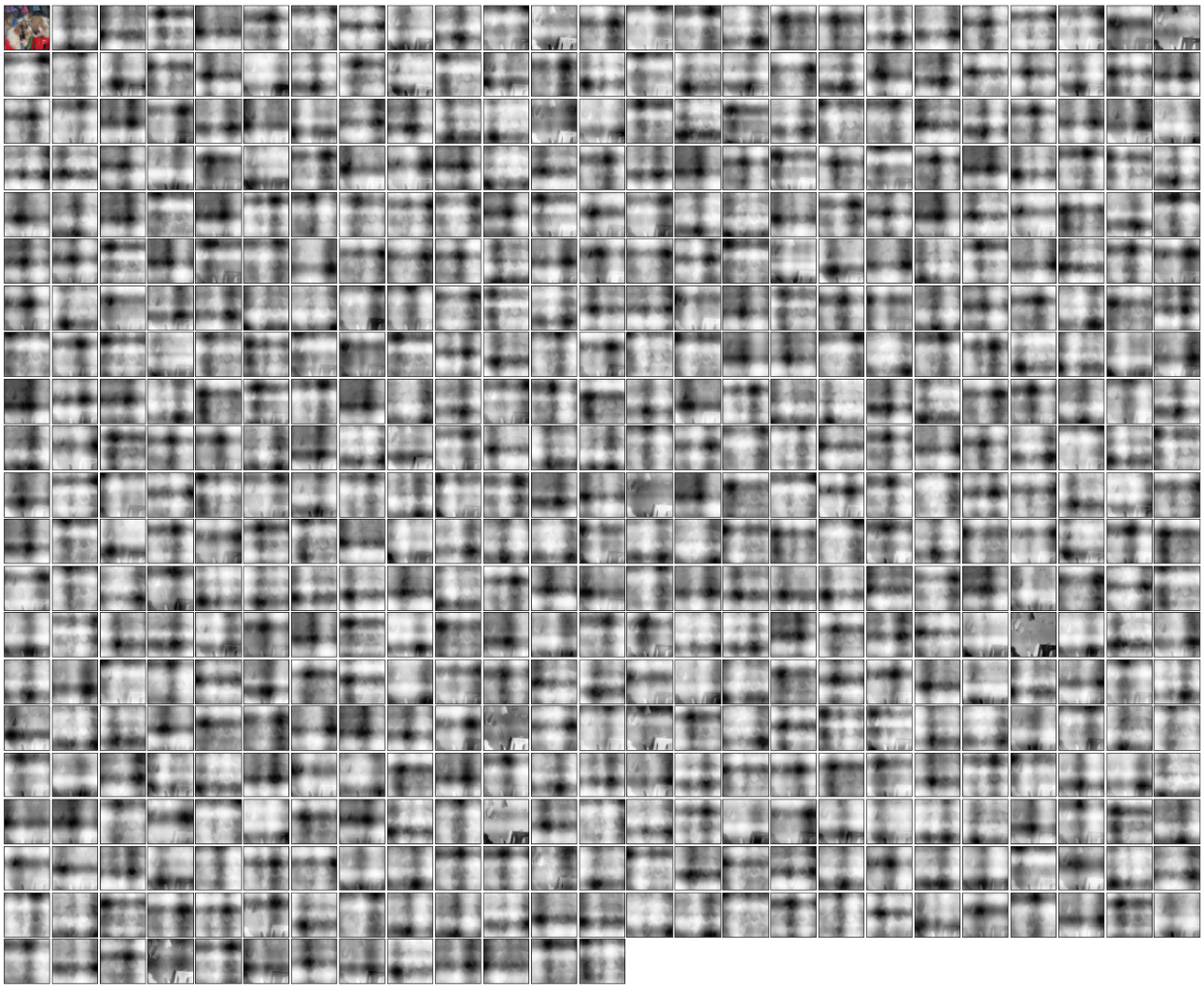


Figure 9. Example attention maps from the **first cross-attend** of an ImageNet network trained with **2D Fourier feature** position encodings.

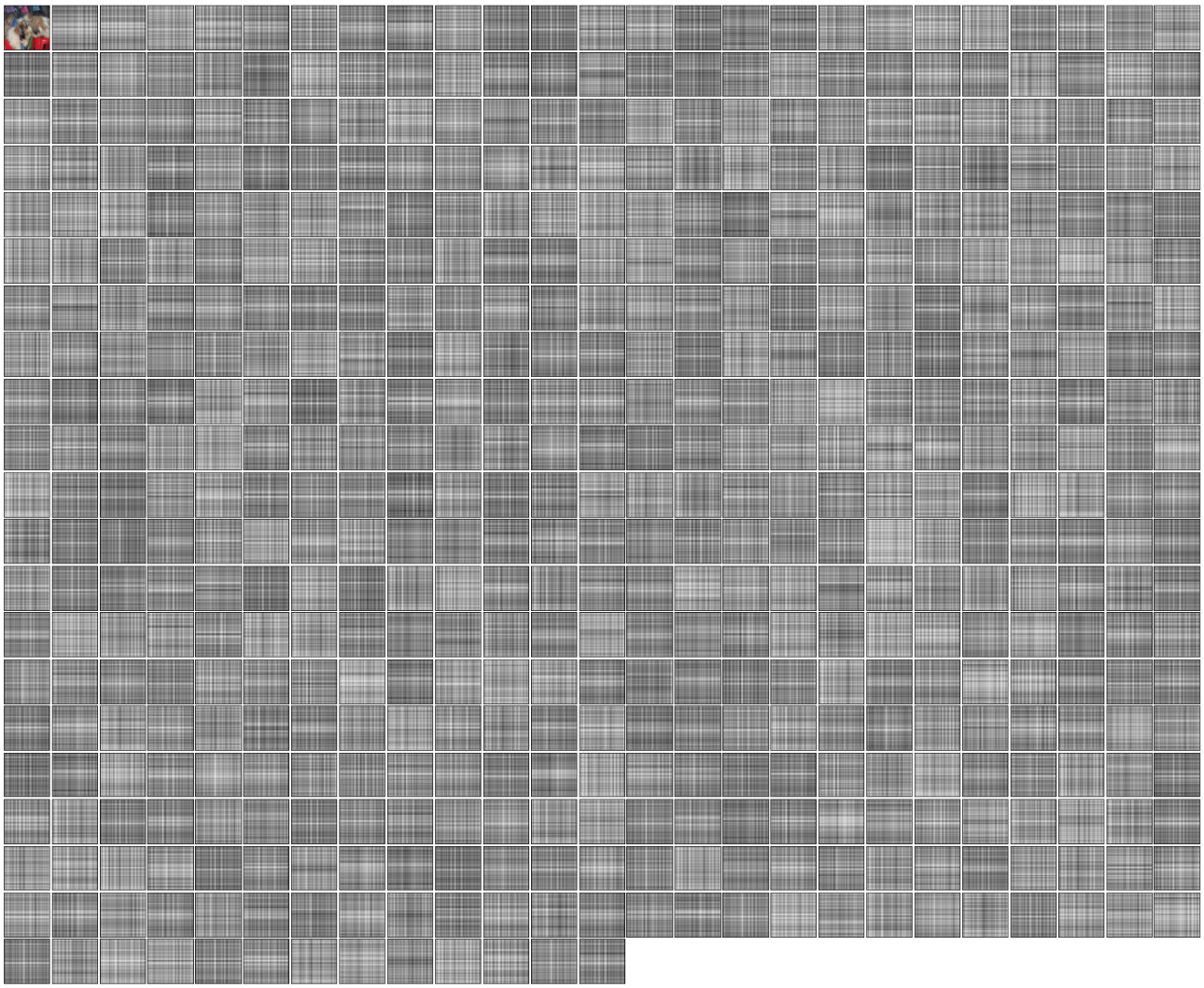


Figure 10. Example attention maps from the **eighth (final) cross-attend** of an ImageNet network trained with **2D Fourier feature** position encodings.

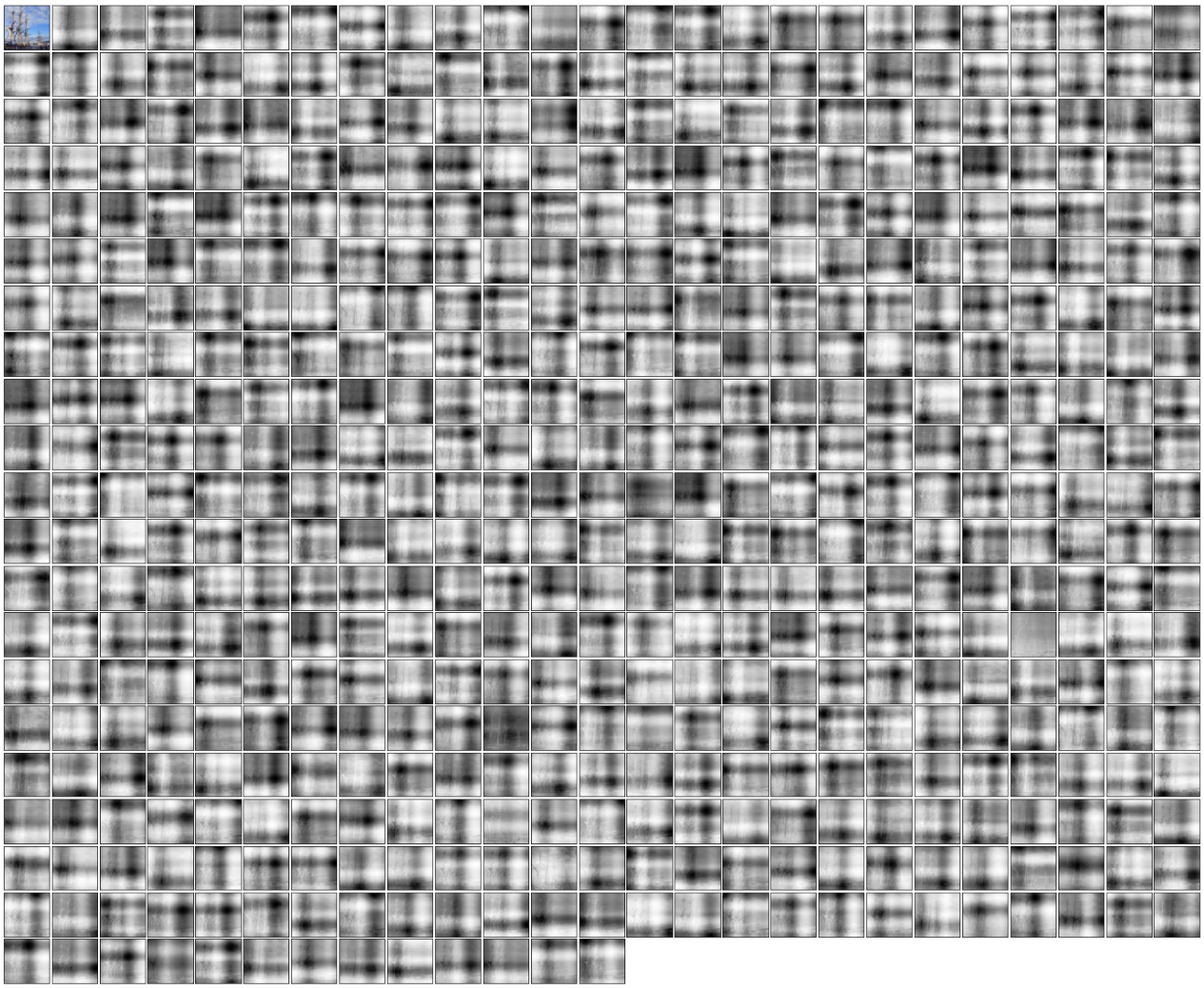


Figure 11. Example attention maps from the **first cross-attend** of an ImageNet network trained with **2D Fourier feature** position encodings.

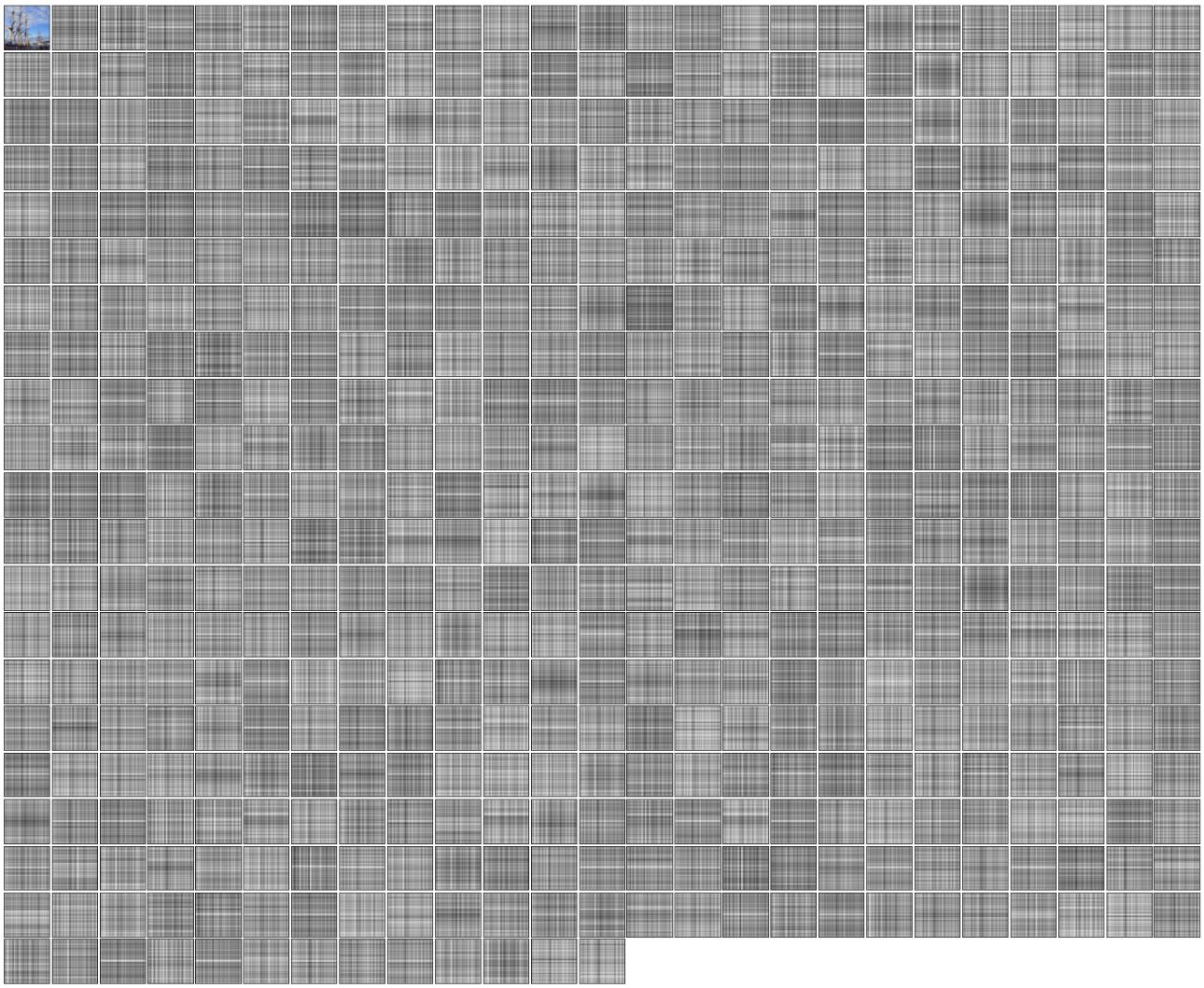


Figure 12. Example attention maps from the **eighth (final) cross-attend** of an ImageNet network trained with **2D Fourier feature** position encodings.

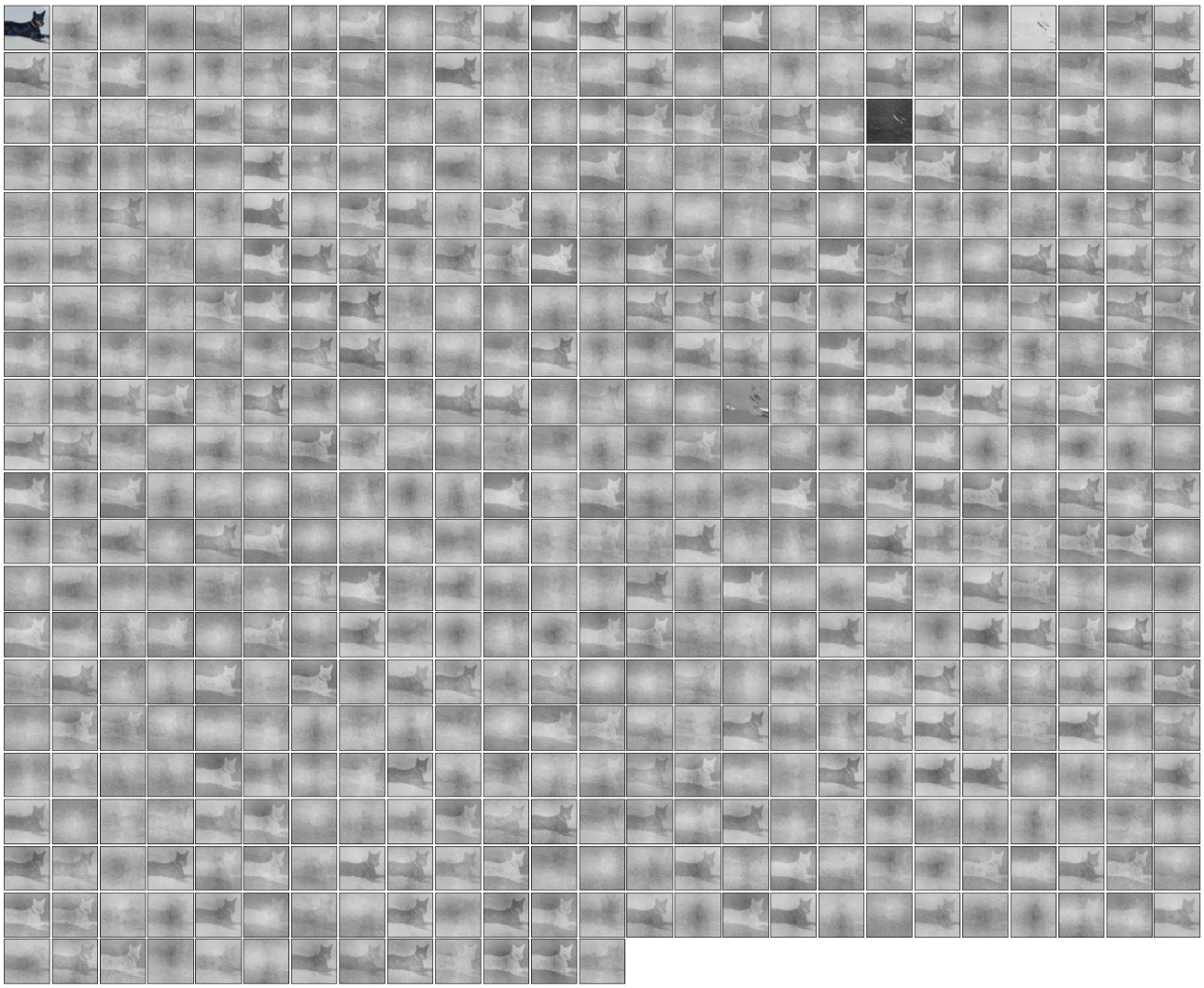


Figure 13. Example attention maps from the **first (only) cross-attention** of an ImageNet network trained with **learned position encodings**.

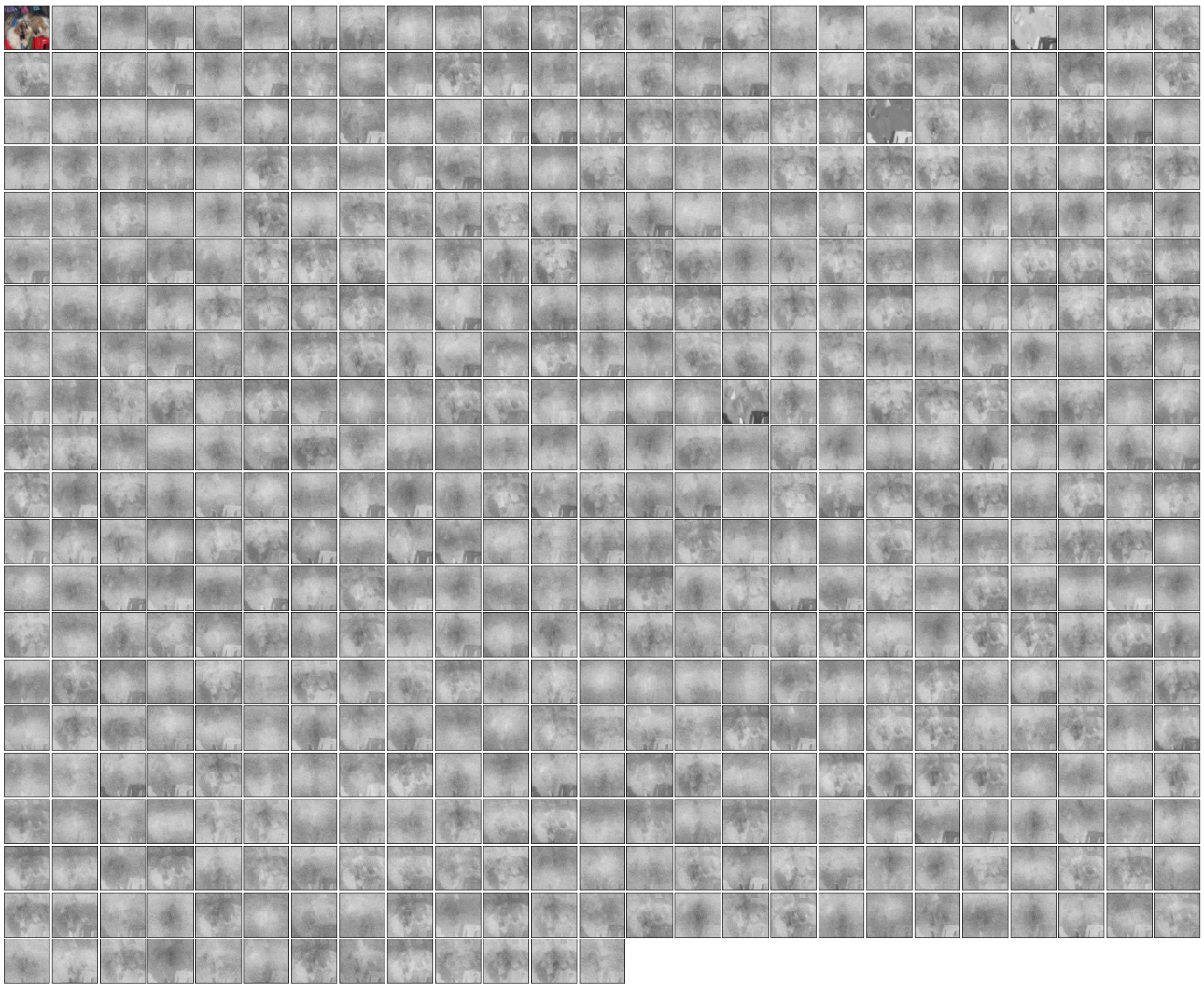


Figure 14. Example attention maps from the **first (only) cross-attention** of an ImageNet network trained with **learned position encodings**.

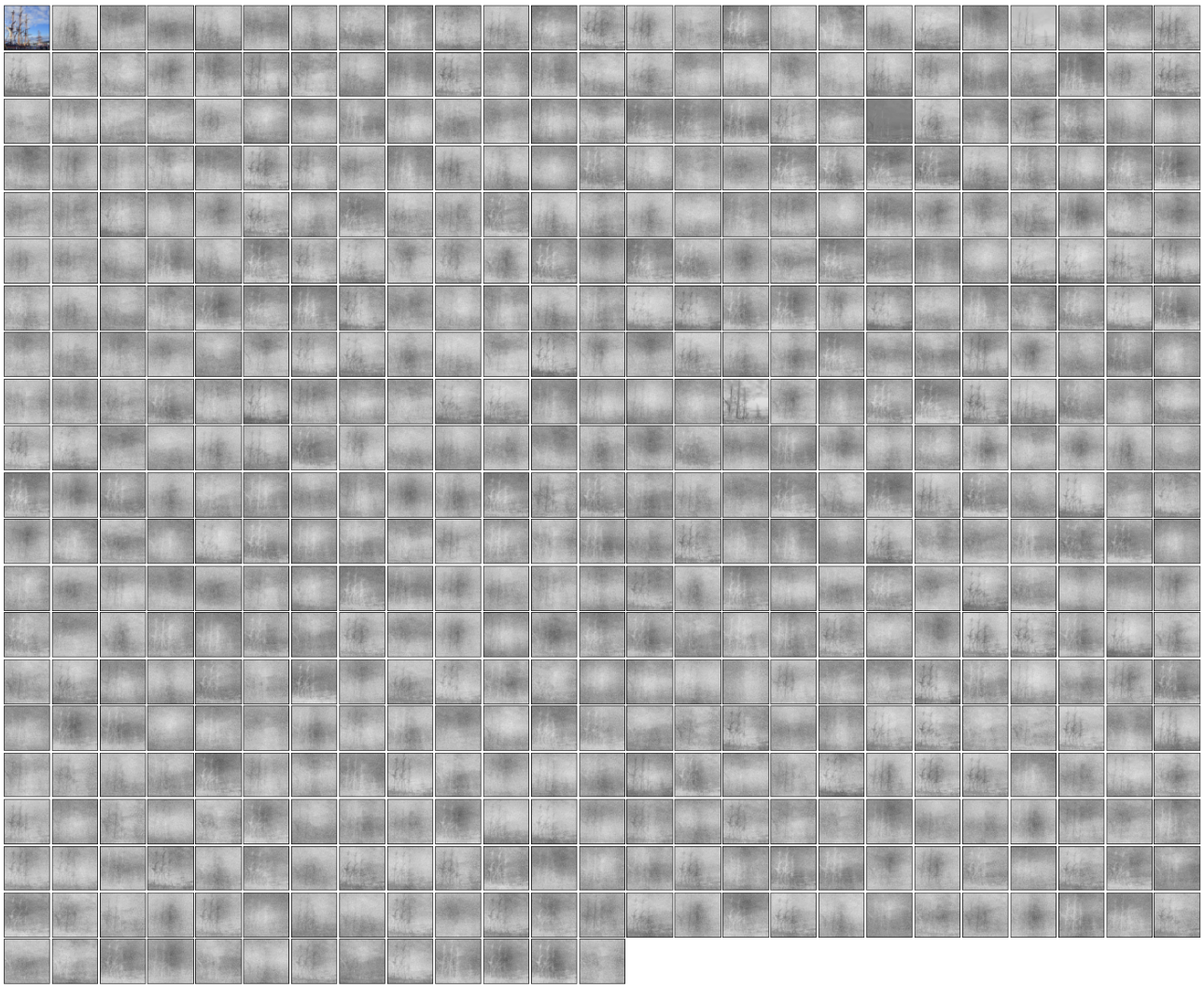


Figure 15. Example attention maps from the **first (only) cross-attention** of an ImageNet network trained with **learned position encodings**.

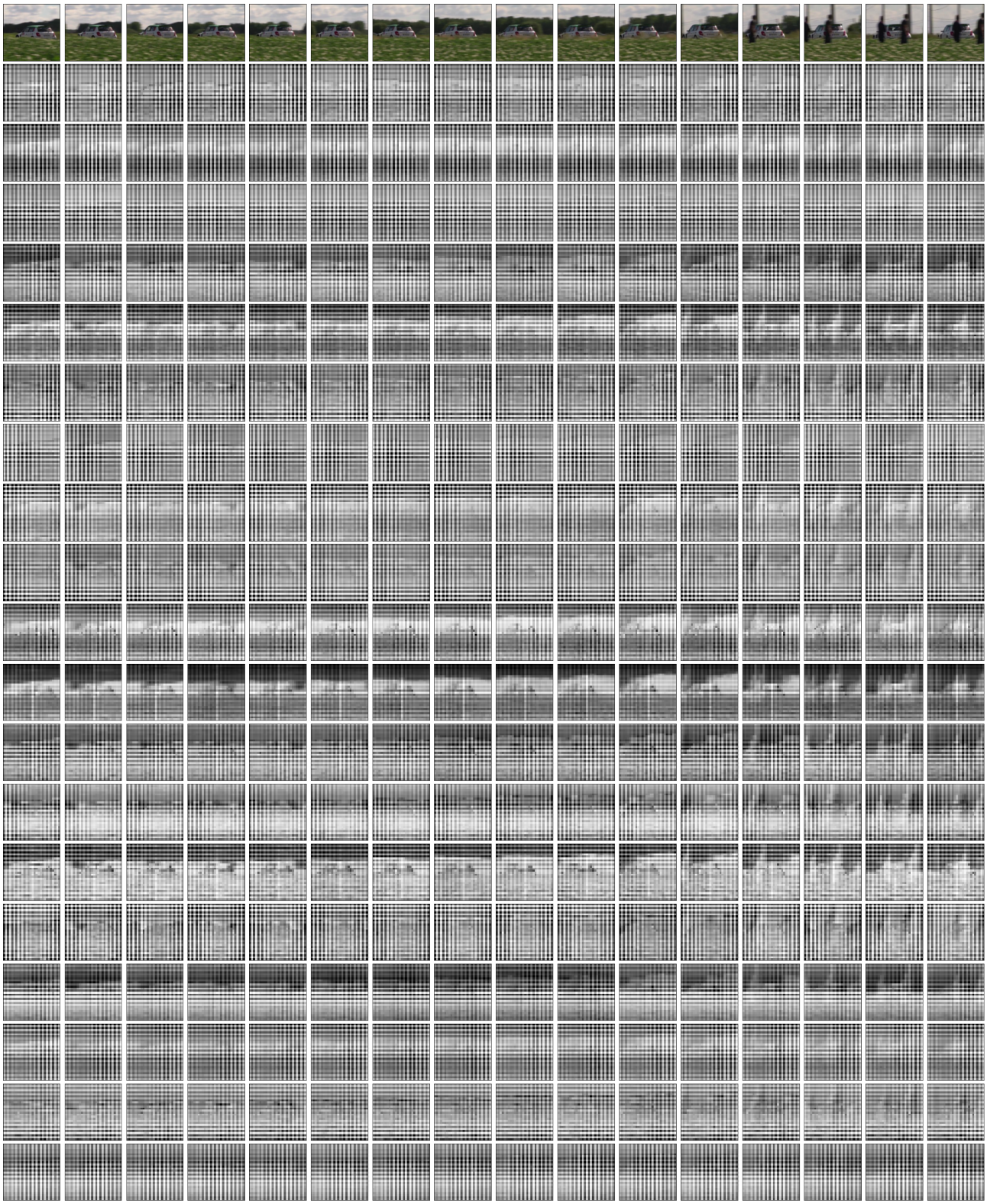


Figure 16. Example attention maps from the **first cross-attend** of an AudioSet network trained on **video only**.

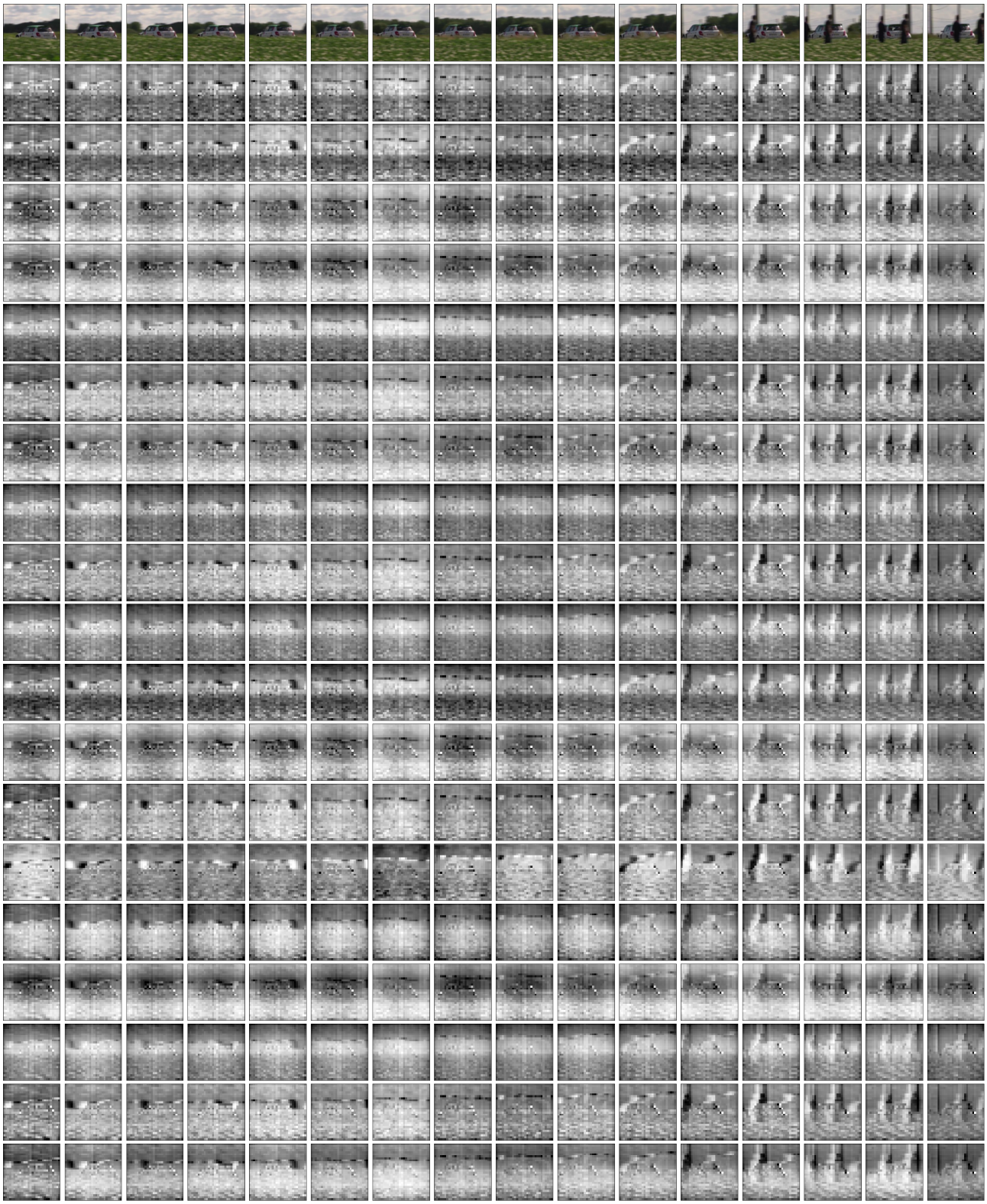


Figure 17. Example attention maps from the **second (final) cross-attend** of an AudioSet network trained on **video only**.

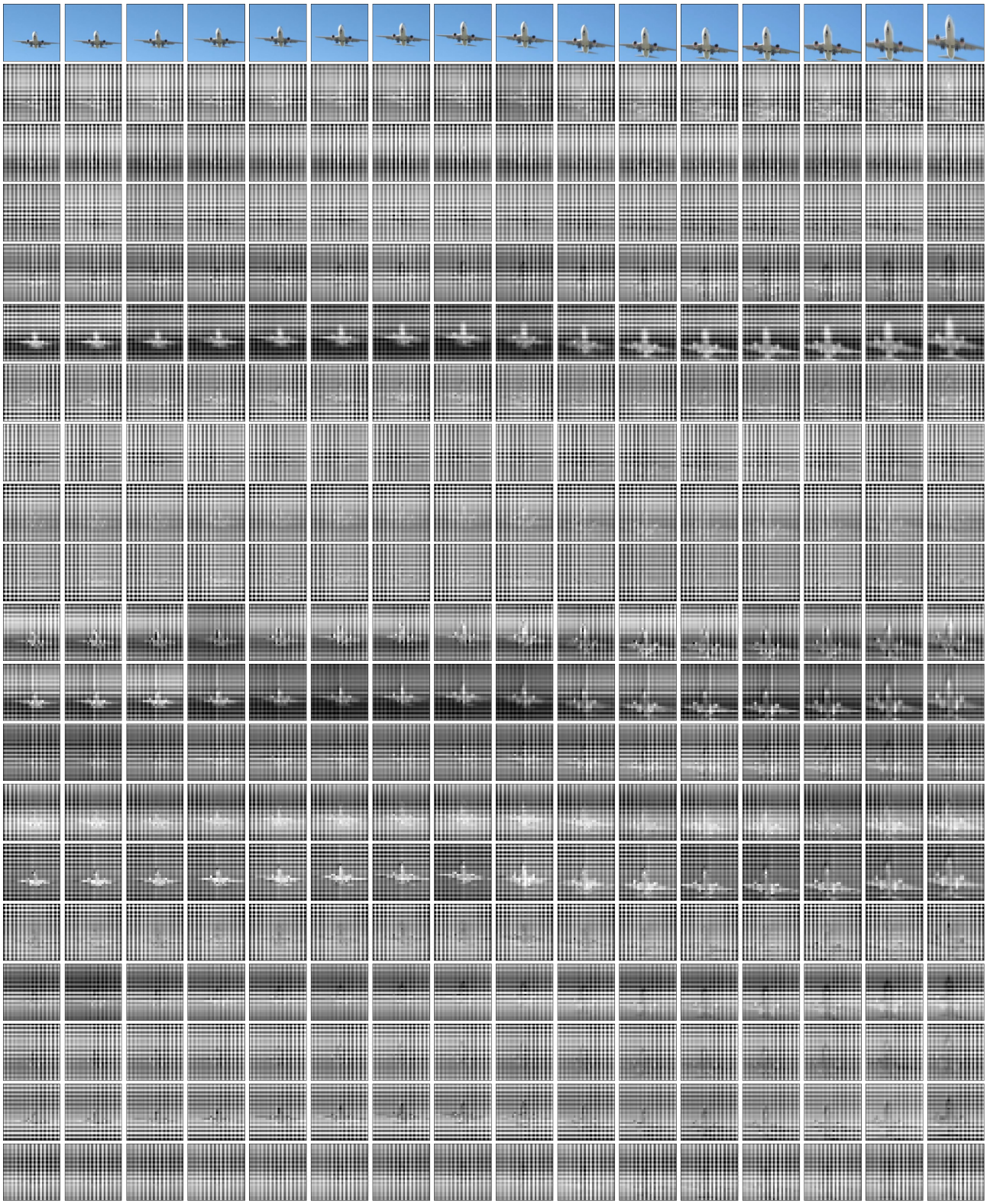


Figure 18. Example attention maps from the **first cross-attend** of an AudioSet network trained on **video only**.

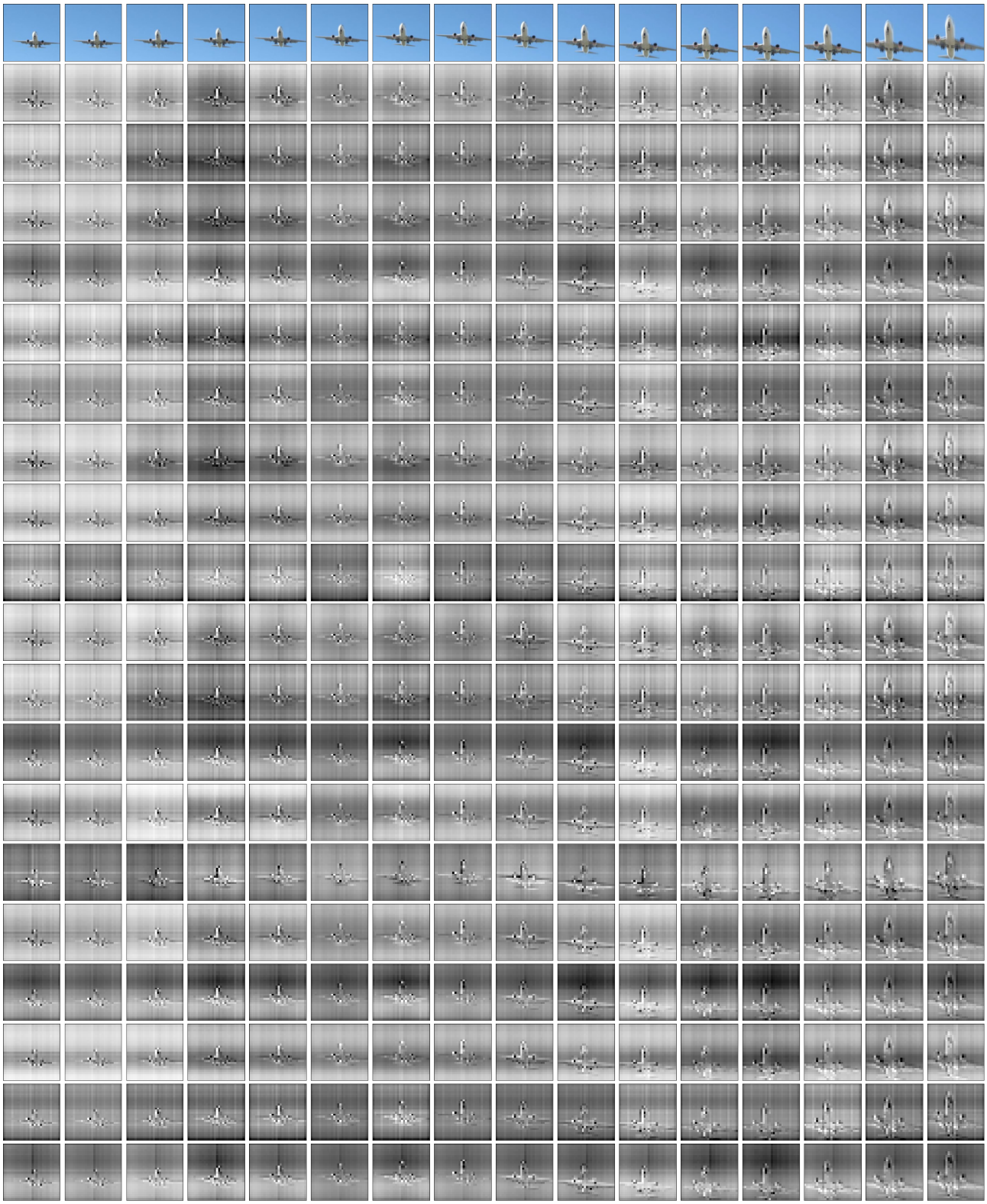


Figure 19. Example attention maps from the **second (final) cross-attend** of an AudioSet network trained on **video only**.

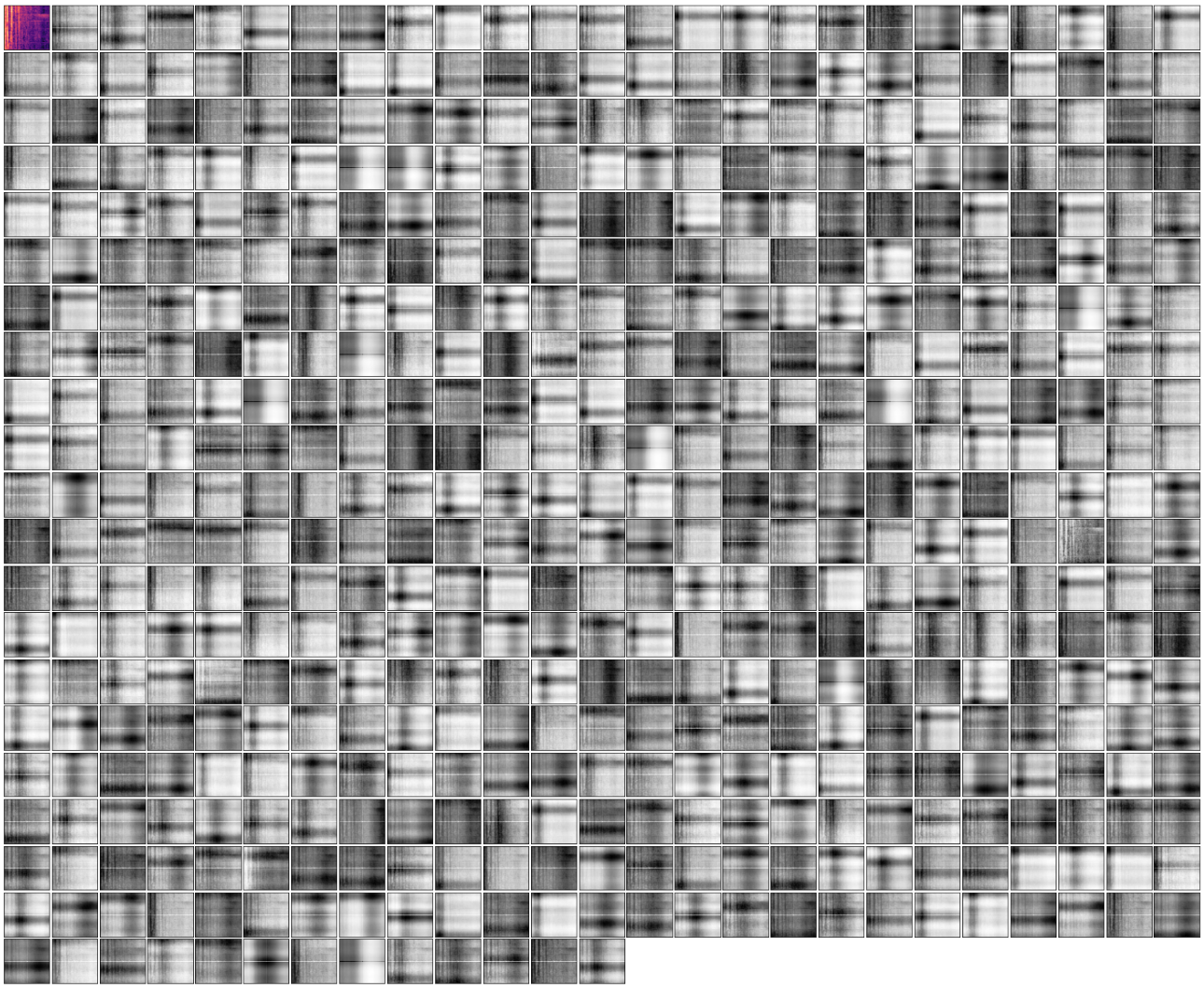


Figure 20. Example attention maps from the **first cross-attend** of an AudioSet network trained on **mel-spectrogram only** (car).

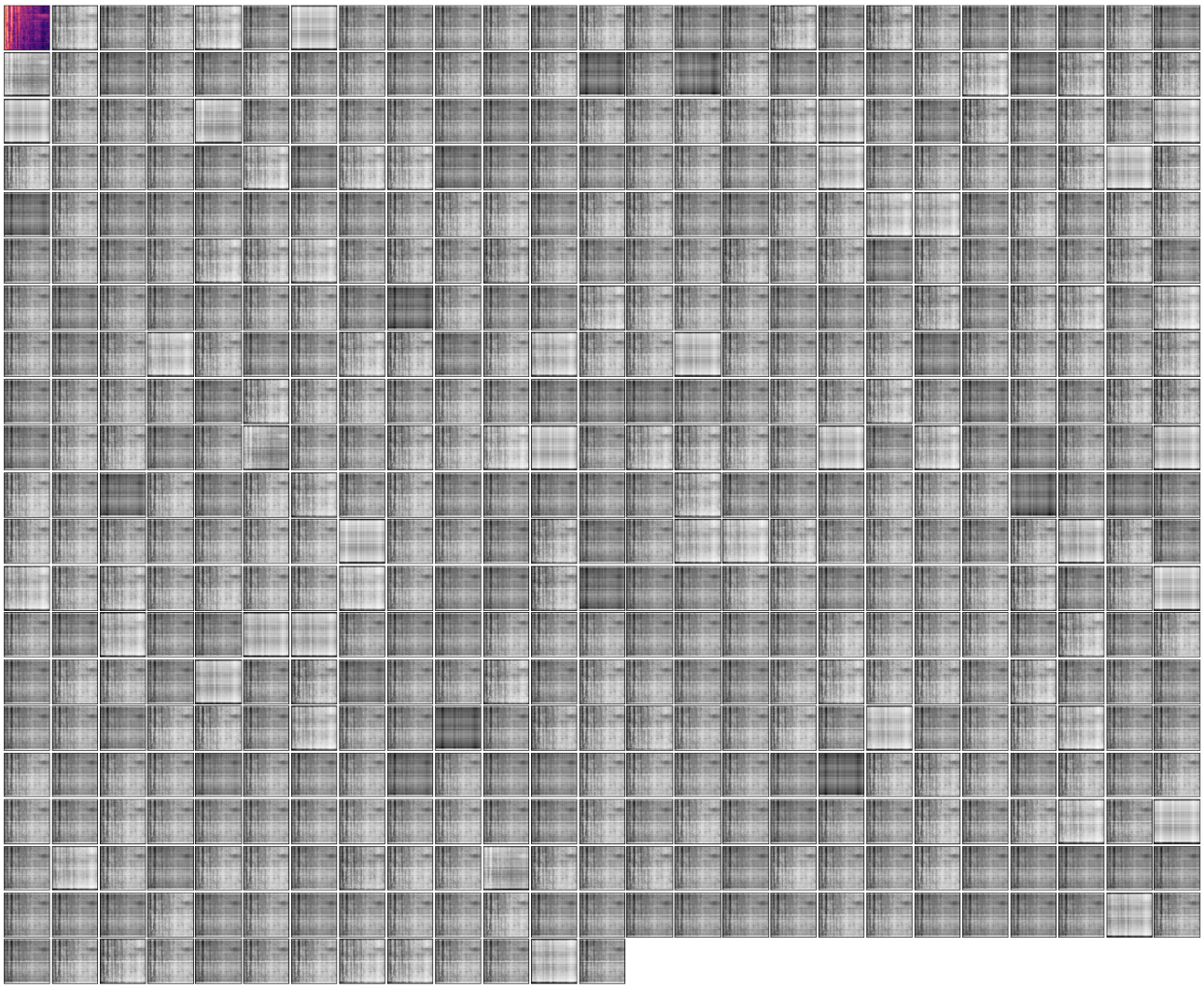


Figure 21. Example attention maps from the **second (final) cross-attend** of an AudioSet network trained on **mel-spectrogram only** (car).

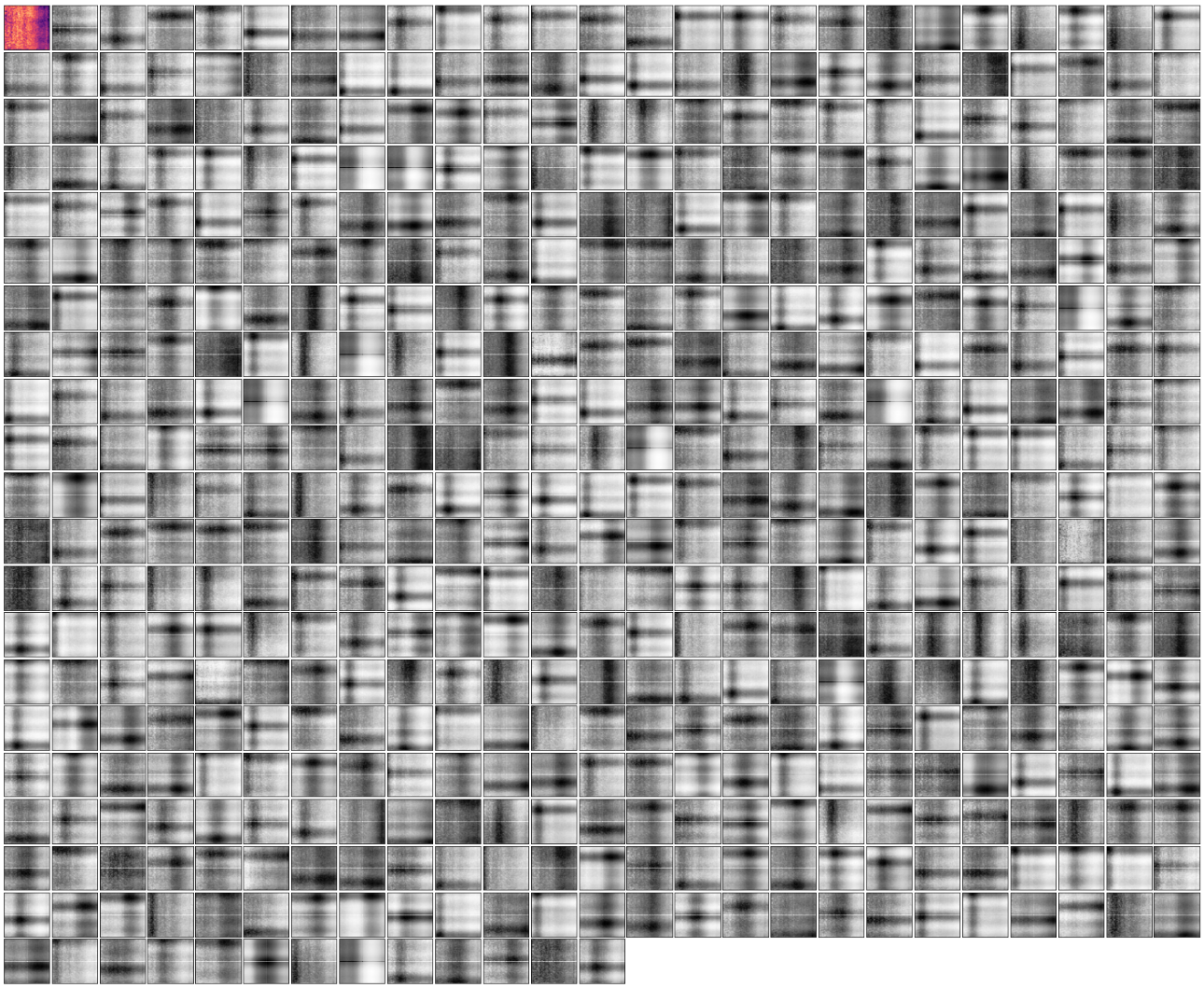


Figure 22. Example attention maps from the **first cross-attend** of an AudioSet network trained on **mel-spectrogram only** (plane).

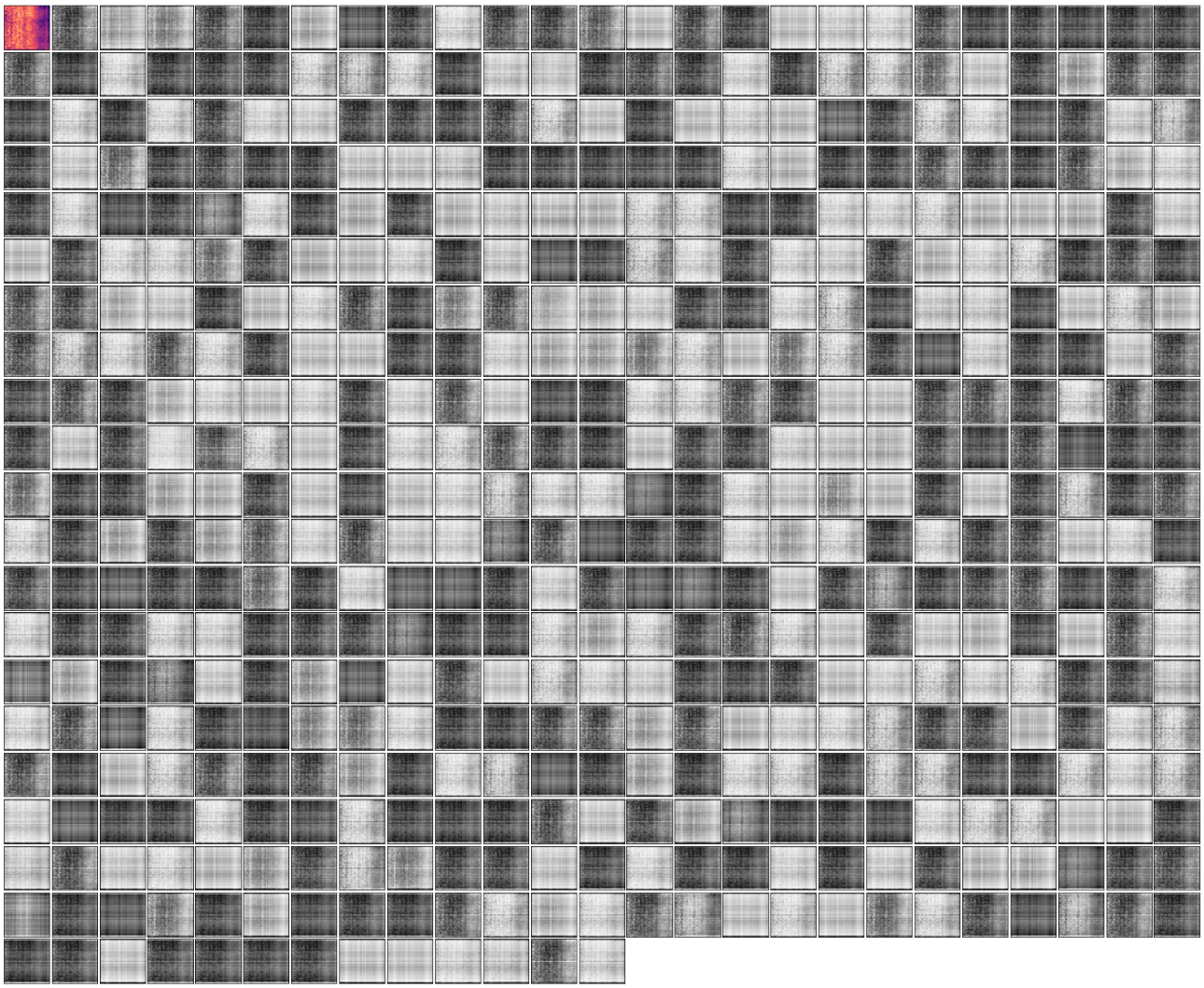


Figure 23. Example attention maps from the **second (final) cross-attend** of an AudioSet network trained on **mel-spectrogram only** (plane).

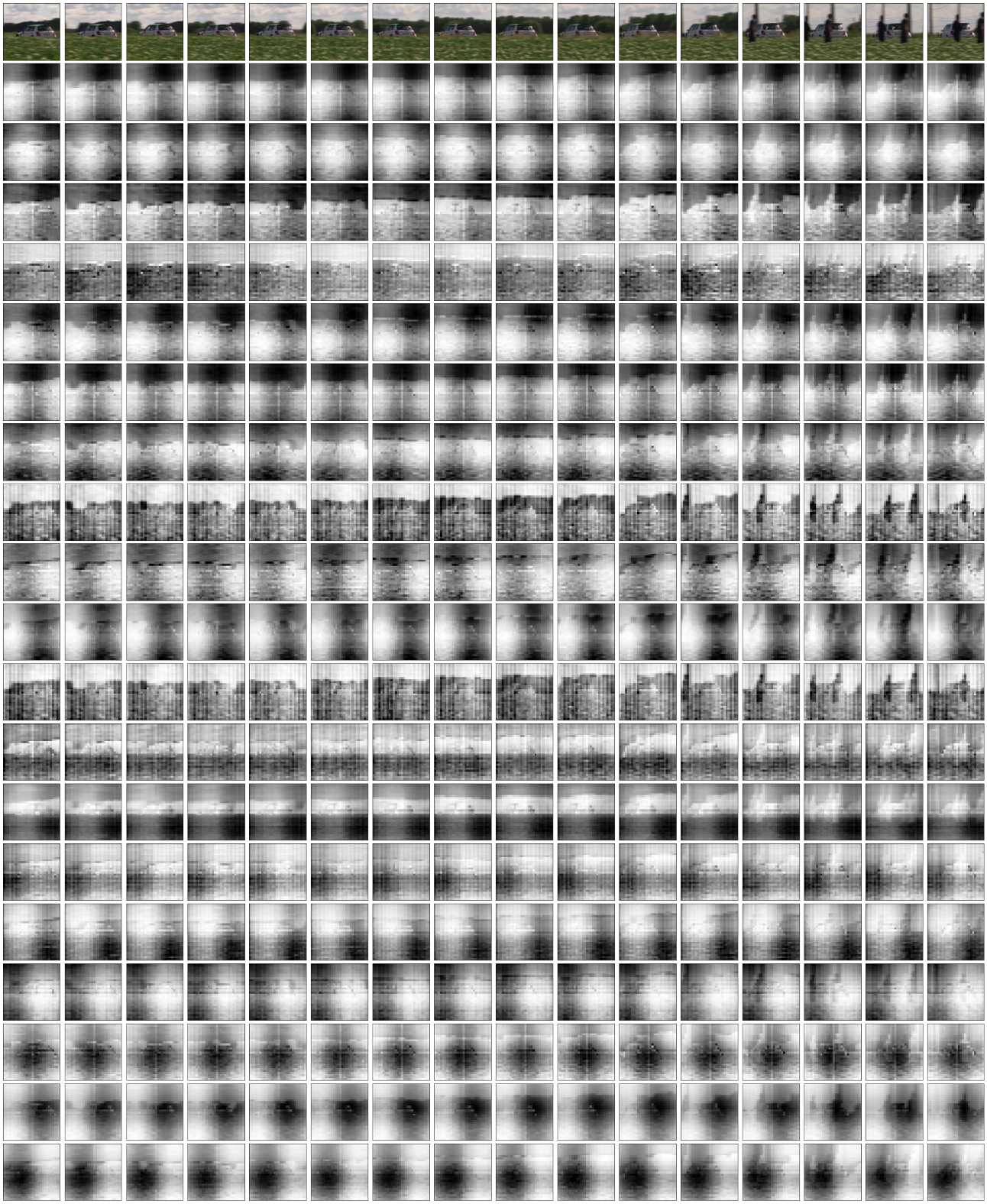


Figure 24. Example attention maps from the **first cross-attend** over the video input subset of an AudioSet network trained on **video and mel-spectrogram**.

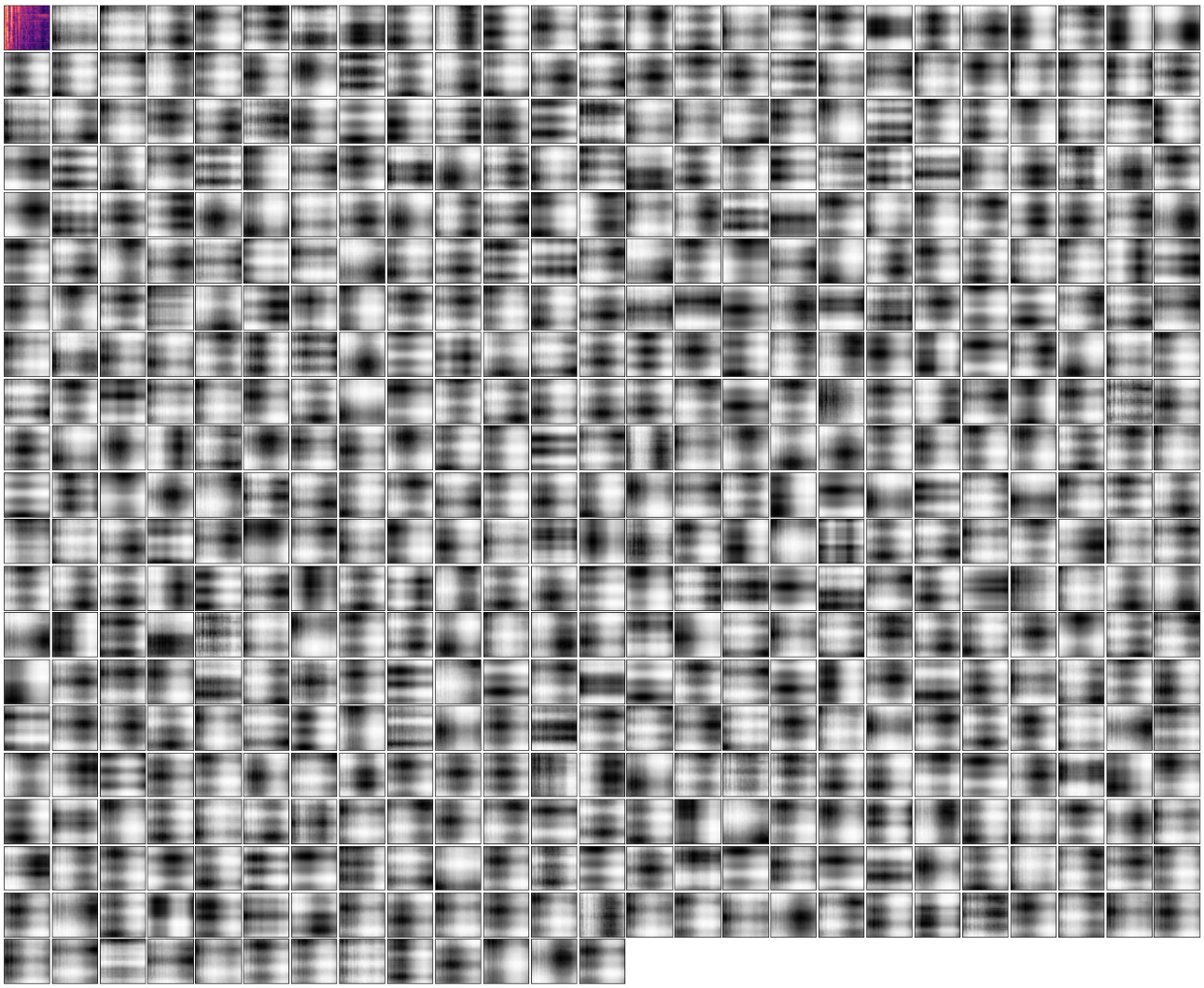


Figure 25. Example attention maps from the **first cross-attend** over the mel-spectrogram input subset of an AudioSet network trained on **video and mel-spectrogram** (car).

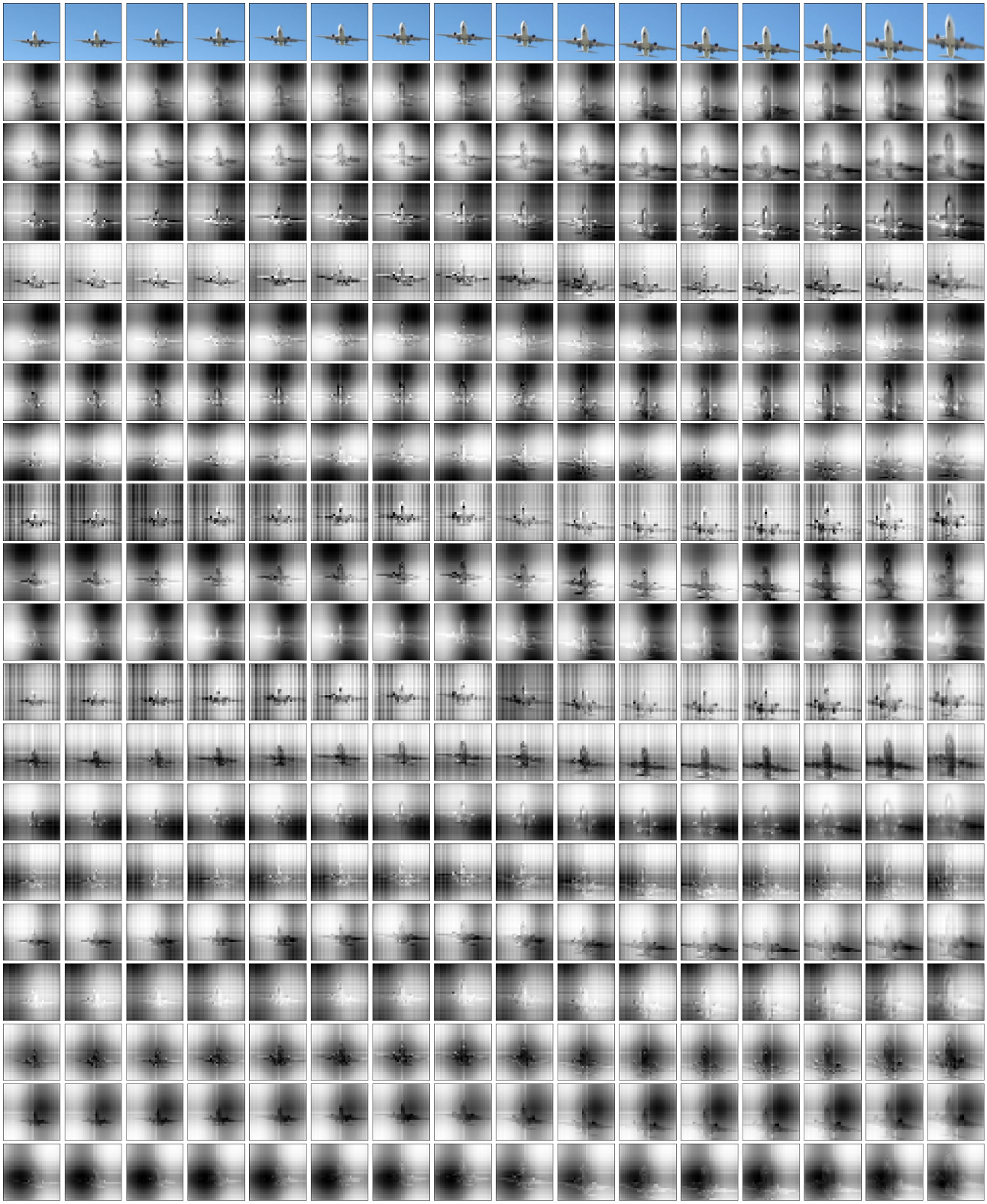


Figure 26. Example attention maps from the **first cross-attend** over the video input subset of an AudioSet network trained on **video and mel-spectrogram**.

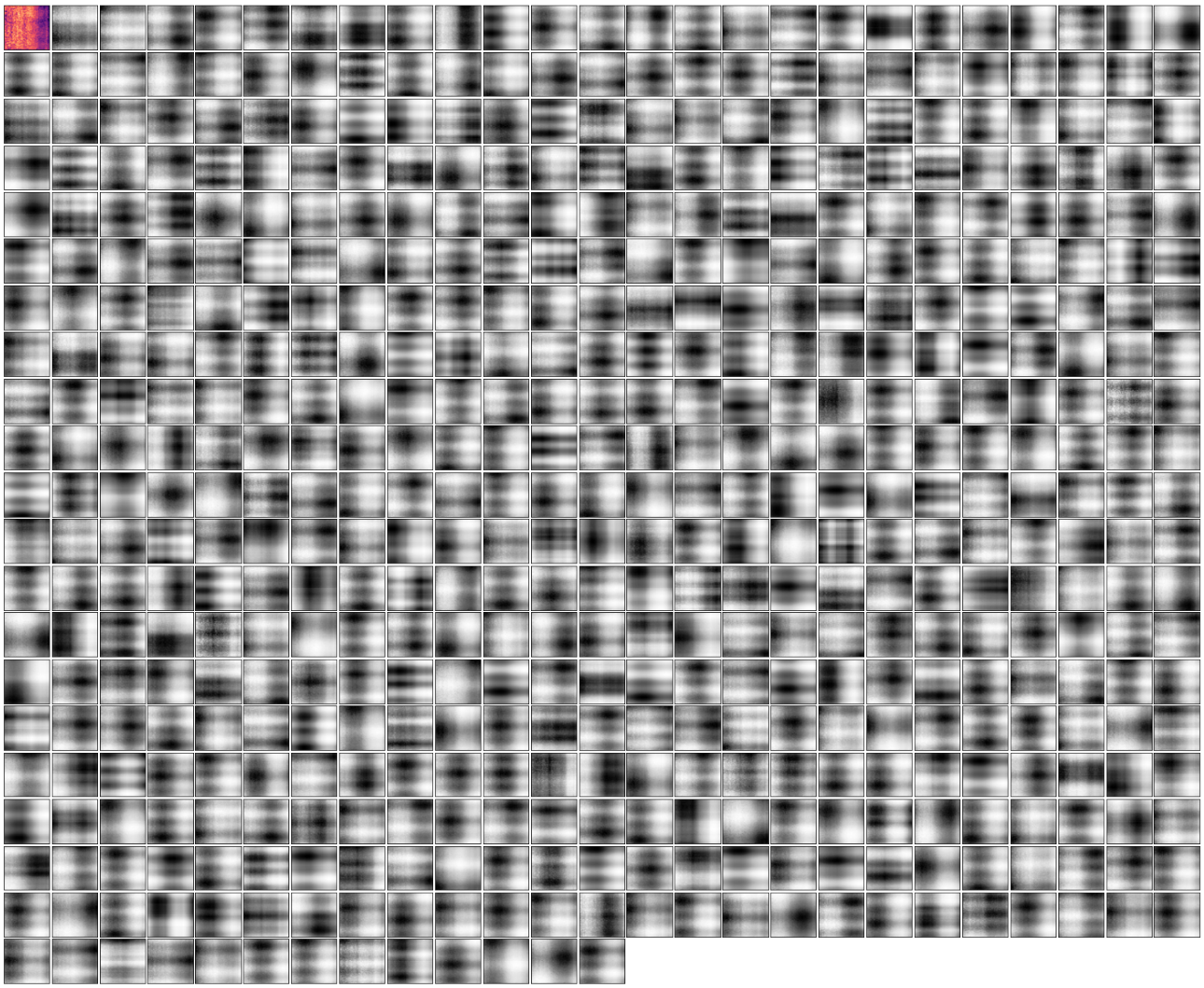


Figure 27. Example attention maps from the **first cross-attend** over the mel-spectrogram input subset of an AudioSet network trained on **video and mel-spectrogram** (plane).

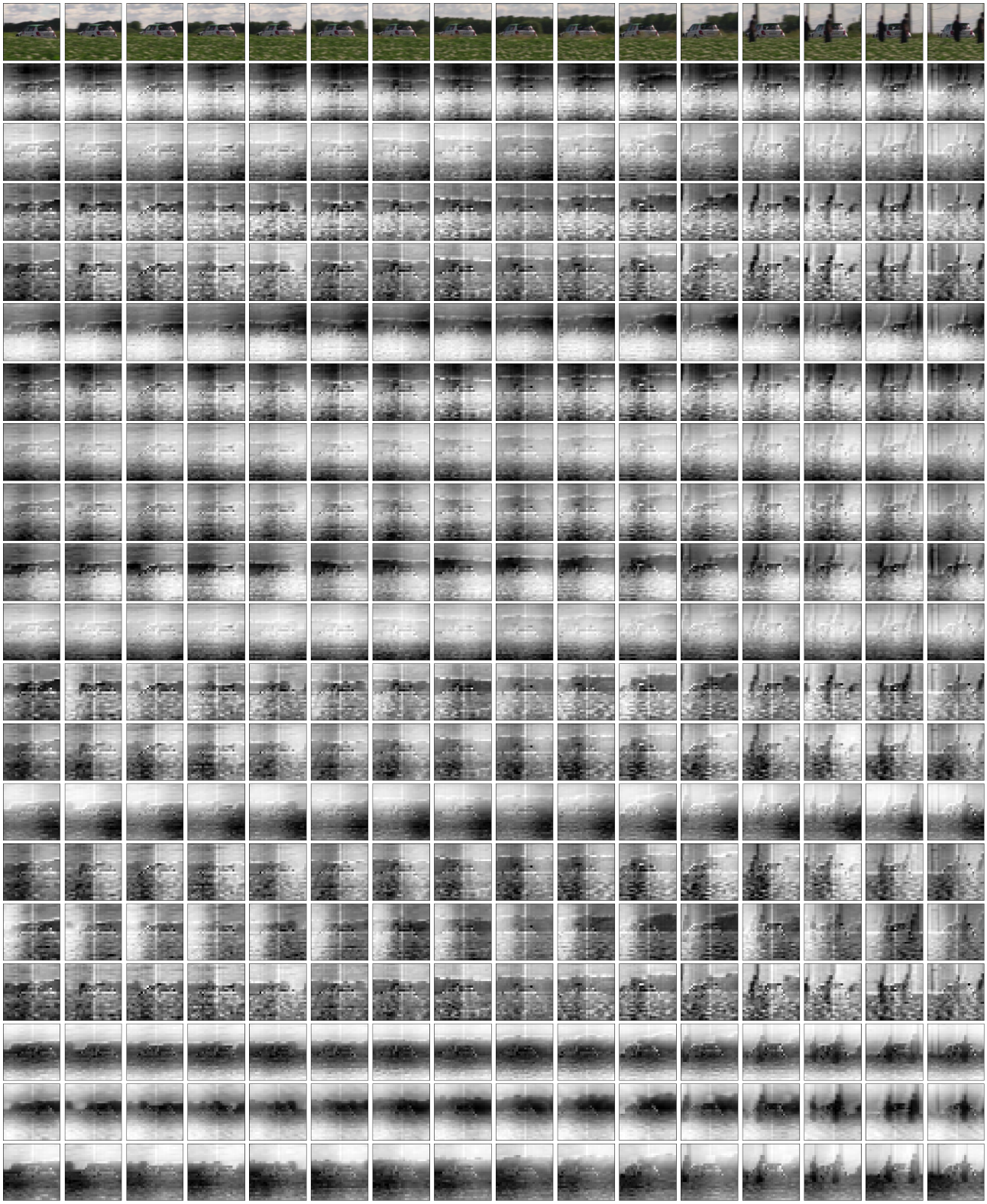


Figure 28. Example attention maps from the **second (final) cross-attend** over the video input subset of an AudioSet network trained on **video and mel-spectrogram**.

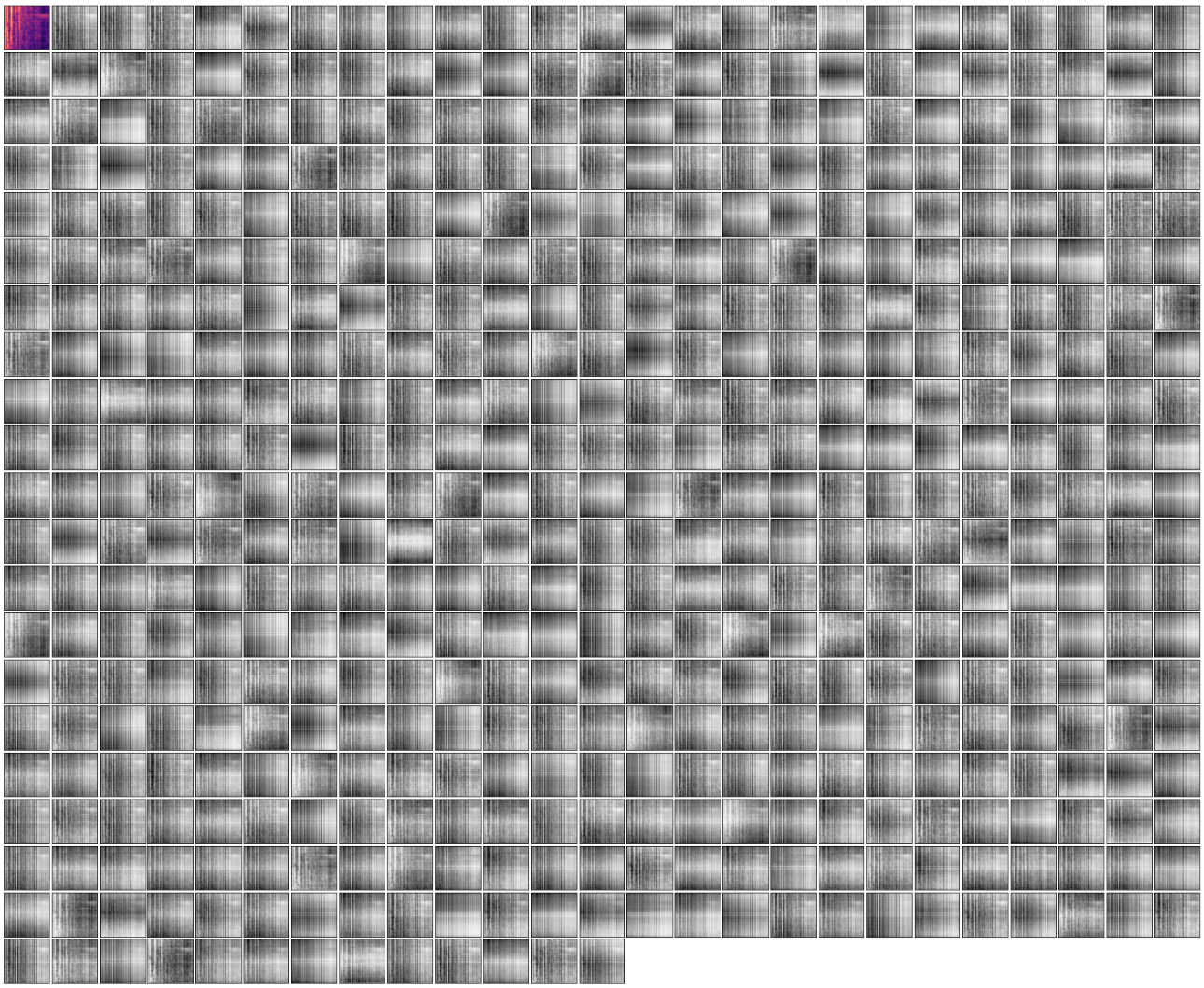


Figure 29. Example attention maps from the **second (final) cross-attend** over the mel-spectrogram input subset of an AudioSet network trained on **video and mel-spectrogram** (car).

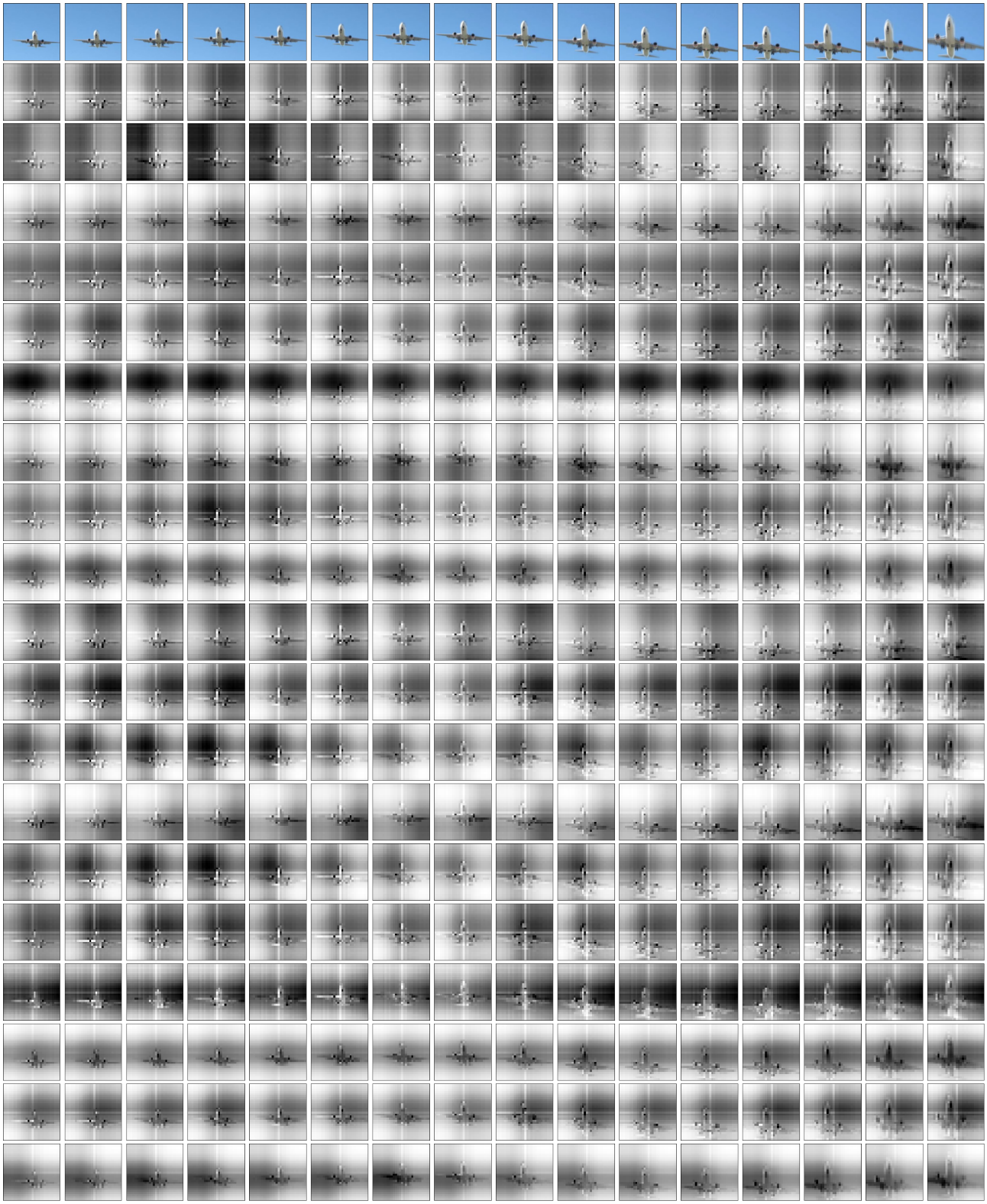


Figure 30. Example attention maps from the **second (final) cross-attend** over the video input subset of an AudioSet network trained on **video and mel-spectrogram**.

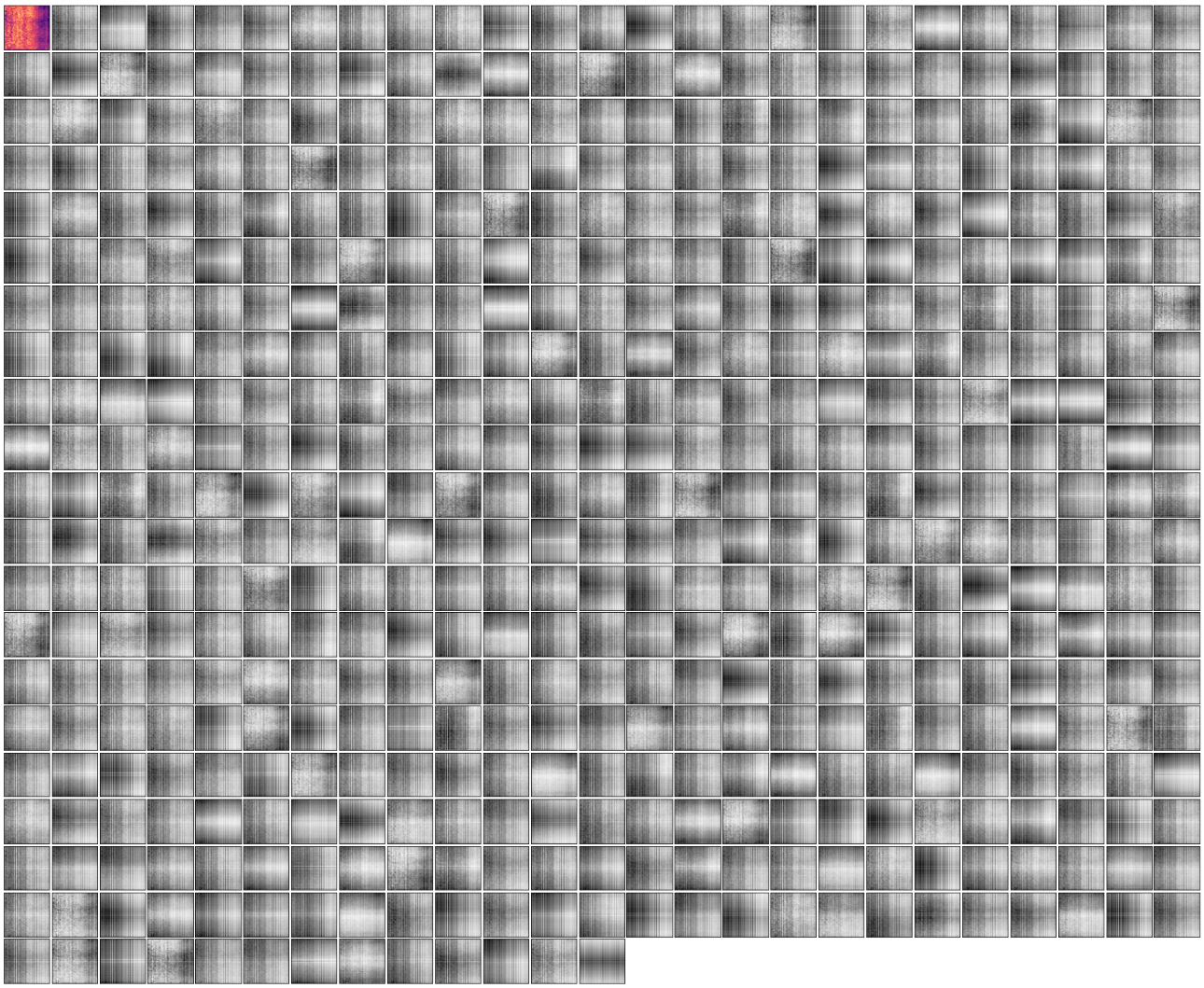


Figure 31. Example attention maps from the **second (final) cross-attend** over the mel-spectrogram input subset of an AudioSet network trained on **video and mel-spectrogram** (plane).