Hierarchical Perceiver

João Carreira 1 Skanda Koppula 1 Daniel Zoran 1 Adria Recasens 1 Catalin Ionescu 1 Olivier Hénaff 1 Evan Shelhamer 1 Relja Arandjelović 1 Matt Botvinick 1 Oriol Vinyals 1 Karen Simonyan 1 Andrew Zisserman 1 Andrew Jaegle 1

Abstract

General perception systems such as Perceivers can process arbitrary modalities in any combination and are able to handle up to a few hundred thousand inputs. They achieve this generality by exclusively using global attention operations. This however hinders them from scaling up to the inputs sizes required to process raw high-resolution images or video. In this paper, we show that some degree of locality can be introduced back into these models, greatly improving their efficiency while preserving their generality. To scale them further, we introduce a self-supervised approach that enables learning dense low-dimensional positional embeddings for very large signals. We call the resulting model a Hierarchical Perceiver (HiP). HiP retains the ability to process arbitrary modalities, but now at higher-resolution and without any specialized preprocessing, improving over flat Perceivers in both efficiency and accuracy on the ImageNet, Audioset and PASCAL VOC datasets.

1. Introduction

Perceiver (Jaegle et al., 2021; 2022) is a recently proposed model that produces competitive results in classic vision tasks like optical flow and ImageNet classification, in text benchmarks like GLUE, in multimodal domains such as audio-visual recognition and auto-encoding, on point cloud data or even in StarCraft agents. All with the same basic architecture and without the need for any data preprocessing for up to 100k inputs / outputs.

Perceivers build upon Transformers (Vaswani et al., 2017) but introduce an efficient global attention mechanism that allows them to handle much larger input/output spaces. They operate on a matrix of flattened inputs – where rows correspond to pixels, raw audio time steps, characters, etc., and columns correspond to features, such as RGB values. Perceivers produce a flat matrix of outputs controlled by an input query matrix (Jaegle et al., 2022). Prior information, such as spatial coordinates, is featurized – typically using Fourier features for images and audio data – and concatenated with the inputs and the queries.

In this paper we introduce Hierarchical Perceiver (HiP), a model that builds upon Perceivers, but is faster, scales to even more inputs/outputs, reduces the need for input engineering and is also more accurate. These improvements stem from two ideas. First, we build back locality into the architecture while preserving its modality-independence, by leveraging the observation that intrinsic locality in input and target output signals is often preserved after even simple, generic flattening operations, as illustrated in Figure 1. Second, we show that random masked auto-encoding (MAE) enables us to learn low-dimensional positional embeddings for high-resolution signals such as images or audio, doing away with the need for externally-defined, hand-designed Fourier embeddings, which are also computationally expensive. Previous papers also experimented with learned positional embeddings, but reported poor results when just training for supervised classification tasks.

We describe HiP’s architecture in Section 3, then analyze and ablate the model on the ImageNet dataset, showing that it is faster and more accurate than its predecessors. We also show that learned low-dimensional positional embeddings are impoverished and underperform if trained with a supervised classification loss, but are rich and slightly better than high-dimensional Fourier embeddings if trained with random pixel masked auto-encoding. We also report competitive results on AudioSet using raw audio and video, without patching or convolutions (we do not use those in this paper). Finally, we report results for semantic segmentation on PASCAL VOC and analyze the versatility of the model on ModelNet40 point cloud classification. The paper concludes with a discussion of connections of this work to other areas and current limitations.
2. Related Work

Computer perception is a research area that seeks to develop flexible models that can solve tasks in images, video, audio, and many other modalities. We outline work on computer perception that feeds into the Hierarchical Perceiver.

2.1. Attention and general-purpose architectures

Transformers (Vaswani et al., 2017) provide a convenient building block for general-purpose architectures because they do not make strong assumptions about the structure of the data, unlike ConvNets (LeCun et al., 1998). Transformers, however, have a well known quadratic growth in complexity with the number of inputs.

ViT-based models (Dosovitskiy et al., 2021) have been applied to multimodal inputs (Akbari et al., 2021; Nagrani et al., 2021; Likhosherstov et al., 2021). They deal with the problem of large input spaces by preprocessing the inputs into a small number of high-dimensional vectors using modality-specific operations or architectures.

Perceivers (Jaegle et al., 2021) can operate on raw inputs by adapting Transformers to operate on a latent bottleneck space, making it possible to deal with significantly larger input and output spaces and to build very deep models. Perceivers demonstrated competitive results in various domains, including audio, text and images (Jaegle et al., 2022). However, they still require domain-specific preprocessing such as patching or convolutions for very large signals like high-resolution images or videos.

2.2. Hierarchical models

Unlike the equally popular ConvNets, most Transformer-based models are flat, having a constant number of activations and channels throughout. Hierarchical variations have been recently proposed for NLP (Dai et al., 2020; Nawrot et al., 2021) and for vision, building upon ViT (Fan et al., 2021; Liu et al., 2021; Ranftl et al., 2021). Here we propose a hierarchical version of Perceiver that reuses its technique of cross-attending to learned latents, but now also for adapting the internal resolution and number of channels between layers.

2.3. Positional embeddings

Transformers encode positional information by summing or concatenating specialized, hardcoded embeddings (Vaswani et al., 2017; Devlin et al., 2019; Jaegle et al., 2021) (e.g. sinusoidal, Fourier, etc.), instead of defining and operating on local neighborhoods as in convolution-based architectures. While some prior works demonstrate learned embeddings on patched inputs (e.g. Dosovitskiy et al. 2021), to the best of our knowledge, scaling low-dimensional, completely-learned position embeddings to a million and more inputs has remained an open challenge.

2.4. Masked auto-encoding

The combination of masked auto-encoding (MAE) (Vincent et al., 2008) and Transformers has been tremendously successful in natural language processing, pioneered by famous models like BERT (Devlin et al., 2019) and RoBERTa (Liu et al., 2019). This success has recently started to percolate into specific perception domains: computer vision (He et al., 2021; Bao et al., 2022), point-cloud understanding (Yan et al., 2022), and speech recognition (Hsu et al., 2021). None of these previous works demonstrated, however, the impact of masked auto-encoding in learning high-resolution, modality-agnostic positional embeddings as done in this paper.

2.5. Architectures for high-resolution data

Modern architectures generally handle high-resolution data through convolutions and pooling (e.g. Wang et al. 2020; Ronneberger et al. 2015; Fischer et al. 2015), or through operations that are effectively a convolution (e.g. patching with linear projection, as is the case with Dosovitskiy et al. 2021). More recent architectures like Swin (Liu et al., 2021) similarly implicitly rely on vision-specific image structure to perform windowing and merge operations. HiP uses a non-convolutional approach to tackle high-resolution data that is more modality agnostic.
3. Hierarchical Perceiver

The Perceiver model assumes that input data $X \in \mathbb{R}^{M \times C}$ always arrives as a set of $M$ tokens, each with $C$ channels. If there are different modalities and data sources, pre-processing steps such as flattening and padding may be required to arrive at this form.

The main trick Perceivers use to scale is introducing a set of learned latent vectors $Z \in \mathbb{R}^{K \times D}$ which are updated by cross-attending over the inputs $X$. Expressed as a function of index dimensionality, the complexity of this input encoding process is $O(MK)$ (compared with $O(M^2)$ for Transformers). The set of latents $Z$ can be further processed using $L$ layers self-attention at complexity $O(K^2)$ each, independently of input length. The same scaling trick can also be used to efficiently produce outputs (Jaegle et al., 2022).

Introducing hierarchy. Because Perceivers exclusively utilize Transformer-type operations, they are invariant to permutations of the input tokens and are henceagnostic to the structure of inputs before the flattening operation. Here we lean into the fact that flattening operations often preserve useful local structure for many modalities of interest such as images, video, audio and text. The flatten operation sequentially stacks elements from each index dimension (i.e. spatial or ‘token’ dimensions), working backwards from the last index dimension until all elements have been stacked.1 Because neighboring elements along an index dimension are often nearby in space or time, flattening preserves some (but not all) locality structure. See Figure 1 for an illustration of how local structure is preserved when flattening an image.

Note that neighboring elements after concatenation are not always nearby in space or time. For example, neighboring points after concatenation may belong to two separate rows of an image or may belong to the end of one modality array and the beginning of another modality array. This strategy is only able to partially preserve the locality structure present in input array’s after flattening. For this reason, it represents one strategy for minimally preserving locality structure without re-designing the architecture for domains with new spatial, temporal, or modality structures.

Motivated by these observations we allow our model to exploit locality by the introduction of groups.

The basic HiP block. First, in order to utilize local structure to improve efficiency we group the input to the block by splitting it into $G$ groups, each with $M/G$ tokens. Each of these groups is processed independently by a sequence of Perceiver-like operations: first cross attending within each group with a set of $K$ latent vectors (at $O(MK/G)$ complexity) then applying several layers of Transformer-style self-attention and MLPs on the latents that result within each group (overall complexity $O(G K^2)$ per layer). This is done separately within each group separately but with all parameters shared, except for the learned latent vectors which are different and specific to each group.2 The outputs from all groups are then merged along the index dimension to produce the block output $Y \in \mathbb{R}^{(KG) \times D}$ with $KG$ tokens, each with $D$ channels. This output can be used as input for the next block as explained below. See Figure 2 for an illustration of the block structure.

We can organize blocks that follow this structure in a hierarchical manner where each block treats the previous block’s output as input, grouping and processing accordingly. Figure 2 shows an example of such a hierarchy. In order to pro-

---

1This corresponds to reshaping to 1D with NumPy’s reshape operation (Harris et al., 2020).

2This choice is motivated by the fact that in the multimodal case different groups may have different modalities so may benefit from distinct learned latent parameters.
Hierarchical Perceiver

Figure 3. Example masked auto-encoding results on one ImageNet images using 85% masking rate for groupwise-masking. On the left we show the original image, in the middle the corresponding masked image and on the right the outputs of the 16-group model. Note that the masks were shared across groups (groups in HiP-16 for 224x224 images are sequences of 14 consecutive pixel rows), and this is visible as a vertically recurring pattern.

reduce dense outputs we again generalize Perceiver IO (Jaegle et al., 2021) and construct a reverse hierarchy of layers for decoding. For classification we can cross-attend a learned query with the features directly from the bottleneck layer.

**Depth and width.** While Transformer-based models typically have a constant number of channels in all layers, several recent successful models, particularly in vision (Fan et al., 2021; Liu et al., 2021), have adopted a ConvNet-type channel-depth progression – using more channels deeper in the model where “resolution” is lower. This offers efficiency benefits and may provide a useful inductive bias. We follow this principle in HiP as well.

**Skip connections.** We use sum-type skip connections between encoder and decoder before the cross-attention layers, inspired by encoder-decoder vision architectures such as Hourglass or U-Net (Newell et al., 2016; Ronneberger et al., 2015), as these were shown to help with dense labeling tasks. We found them helpful for stabilizing training in masked auto-encoding – especially in audio – and for obtaining slightly lower losses.

**3.1. Learned low-dimensional positional embeddings**

Models like ViT (Dosovitskiy et al., 2021) learn a separate positional embedding for each token (e.g. each patch) and combine it additively with the token representation. This is extremely general and flexible – for images it was shown that it is not even necessary to feed the model xy coordinates; spatial information is inferred by training directly on the classification task. Learned embeddings work well for ViT as the number of inputs is small, e.g. 256, but for Perceivers which can have orders of magnitude more inputs it works poorly (Jaegle et al., 2021; 2022). Instead, what works well with Perceivers is to concatenate hand-designed Fourier positional embeddings with every input.

**Downsides of Fourier embeddings.** There are two main issues with Fourier embeddings. First, they must be tailored to each modality, depending on whether the data is 1D, 2D, or 3D, and with the maximum frequency and number of frequency bands depending on properties of the input signal determined by human expertise or hyperparameter sweeps. The second problem is that they are large and become a memory bottleneck when trying to scale up models to high-resolution data. For example Perceiver IO used 64 frequency bands for ImageNet experiments, which corresponds to 258 fourier features. For a short 32-frames video at 224 x 224 resolution these positional embeddings alone would take 13GB of GPU or TPU memory, making it impractical to train models. ViT’s learned embeddings are even higher-dimensional.

**Self-supervised learning to the rescue.** We will show in the experimental section that it is feasible to learn low-dimensional positional embeddings but that a plain classification loss is not enough to fully develop them. Instead, what is more effective is to train the models using random, uniform masked auto-encoding, before training on sparsely-supervised downstream tasks like classification. We use a separate linear layer for each input modality to map all inputs to the same dimensionality, then add the learned positional embeddings for every input, using either 16 or 32 dimensions.

In more detail: we randomly and uniformly mask a fixed percentage of the input tensor – this is controlled by a hyperparameter. The input tensor can have multiple modalities and we mask them all uniformly with the same probability. Instead of using the classic masking approach (Devlin et al., 2019) of replacing the masked inputs with a learned token, we choose the more efficient alternative of dropping the masked inputs (He et al., 2021), which we refer as groupwise-masking. This is not without limitations though, in our case. Because of the model’s group structure, in order to optimize performance on current hardware we subsam-
| Configuration | HiP-16 | HiP-256 |
|--------------|--------|---------|
| Input channels | 32     | 16      |
| Groups       | 16     | 16      |
| Self-attention layers | 2     | 1       |
| Heads        | 4      | 2       |
| Channels     | 128    | 64      |
| Latent vectors per group | 128 | 32      |

Table 1. Hyper-parameters defining the two architecture configurations considered in this paper: HiP-16 and HiP-256. Each number inside the brackets refers to a different block (e.g. block 0, block 1, etc.). See Figure 2 for a diagram showing the structure of a block as well as an overall example configuration. In both cases the number of channels increases closer to the model bottleneck, the number of groups decreases down to 1 and most of the processing happens in the bottleneck – we apply 18 self-attention layers there. The difference between both models is in the number of blocks; HiP-256 has two additional blocks on both encoder and decoder with 256 and 64 groups allowing it to process short raw videos. HiP-16 has only 16 groups in the first and last blocks.

4. Experiments

We start with ImageNet and compare the efficiency and accuracy of HiP and Perceiver IO. We also do most ablations and analysis on ImageNet. Afterwards we cover the multimodal setting with raw audio+video on AudioSet, then experiment on a high-resolution vision task: semantic segmentation in PASCAL VOC. Finally we analyze the robustness of the model to point cloud data on ModelNet40. We use HiP-16 in all experiments except in AudioSet where the number of inputs / outputs is large – we try only HiP-256 there.

For masked auto-encoding pre-training we use the optimization hyperparameters from He et al. 2021 and pre-train for 500 epochs with 85% masking for most experiments because that seems to do slightly better than 75% on ImageNet in preliminary experiments. On Audioset we pre-trained for 1000 epochs with 90% masking. We show example masked auto-encoding validation curves in Appendix Figure 7. For finetuning we provide experimental details in each subsection. Other hyperparameters include using a 4x width expansion factor for MLPs in self-attention blocks, a 1x width expansion factor for cross-attention blocks and stochastic depth (Huang et al., 2016) with 30% drop rate in training.

4.1. ImageNet

We use standard 224x224 input crops for a total of 50,176 pixels. Unlike in masked auto-encoding, for classification experiments the decoder is unused. We finetune models similarly to prior work (Liu et al., 2021) – for 300 epochs, using 20 epochs of warmup, the AdamW optimizer (Loshchilov & Hutter, 2019) and a cosine decay schedule.

Comparison to state-of-the-art. Table 3 compares our models with a (far from exhaustive) list of strong models. HiP-16 gets 81.0% accuracy on ImageNet from pixels, up from 79.0 of Perceiver IO also from pixels. HiP-256 is only slightly worse at 79.9%. When comparing to previous Perceiver models with learned position embeddings, accuracy is much higher. Note that masked auto-encoding pre-training
Table 2. Training steps per second (higher is better) for different input image resolutions (224x224, etc.) on TPUv3, using a batch size of 8 images per core. We compare HiP-16 to the ImageNet version of Perceiver IO (Jaegle et al., 2022) for classification, without using the dense hierarchical decoder. OOM stands for “out of memory”, so no timings could be recorded. Perceiver IO and HiP Fourier versions include the default 258-dimensional (64 bands) Fourier positional embeddings, whereas HiP-16 and HiP-256 use respectively 32d and 16d learned embeddings.

Table 3. Results on ImageNet image classification (top-1 accuracy, higher is better). Results for Perceiver / Perceiver IO baselines are taken from the respective papers (Jaegle et al., 2021; 2022) and are from raw pixels – without patching or convolutions.

Table 4. Analysis of the impact of different factors on HiP ImageNet performance (top-1 accuracy, higher is better).

4.2. AudioSet

In this section we pre-train and fine-tune HiP on audio-visual inputs. This domain is especially challenging for its high dimensionality (both waveform and images are directly fed to the model without pre-processing). In particular, the input is a concatenation of the raw audio and the pixels. We use 8 224 × 224 resolution video frames (sampled at 3.125fps) and the corresponding 2.56s of audio (sampled at 48kHz). This results in 401k input samples for video and 123k samples for the audio. To process this very high-dimensional input we use the HiP-256 architecture and pre-train our models using the masked auto-encoding reconstruction loss on the flattened video+audio signal. These experiments aim to...
Hierarchical Perceiver

Figure 4. HiP-16 list of 32 positional embedding channels, with (left) and without (right) MAE pre-training, in both cases followed by classification finetuning, both on ImageNet. MAE leads to rich low-dimensional positional embeddings that are competitive with high-dimensional hand-designed Fourier embeddings for various downstream tasks. Each image displayed is a single channel of the learned embedding. Without this pre-training the resulting learned embeddings obtain very poor downstream performance. Each of these small images is 224x224 resolution, so this is better seen on a computer screen while zooming significantly in.

Challenge HiP with a large reconstruction problem – note that spatial and temporal coordinates are never provided for any of the inputs and must be inferred. Following Jaegle et al. 2021, we use the AudioSet dataset (Gemmeke et al., 2017) which has 1,879,569 clips (audio+frames) for training and 18,427 in the test set (at the time of submission), which are annotated with a multi-label hierarchy consisting of 527 classes. We fine-tune our pre-trained HiP models on audio-visual multi-label classification of the 527 AudioSet classes using a sigmoid cross-entropy loss.

Table 5 presents the classification mean average precision (mAP) of HiP-256 either on raw audio and on raw audio + raw pixels. For fine-tuning and evaluation we use a similar protocol as in Jaegle et al. 2021. When using only audio, our model outperforms the original Perceiver (Jaegle et al., 2021), which ingests pre-processed tokens (128-d “patches” of raw audio). When combining both modalities, HiP performs close to Perceiver (Jaegle et al., 2021), but ingesting significantly more raw inputs. We fine-tune for 300 epochs using the AdamW optimiser (Loshchilov & Hutter, 2019), with a batch size of 256 and a cosine learning rate schedule with maximum value of $3 \times 10^{-5}$ and 5 epochs of warmup. We experimented with longer masked auto-encoding pre-training and higher masking rates on AudioSet: initial models pre-trained for 300 epochs using 85% masking did about 2% absolute worse than the version with 1000 epochs and 90% masking shown in the table.

Table 5. Comparison on AudioSet using raw audio (A) and video (V). The Perceiver baseline numbers are from the paper (Jaegle et al., 2021) and preprocessed the inputs into audio and video patches (e.g. by concatenating pixels within each 2x8x8 space-time volume into a single vector). With HiP we directly feed in individual pixels and raw audio magnitudes.

| Model  | Modalities | #inputs | mAP  |
|--------|------------|---------|------|
| Perceiver | A+V       | 13,024  | 43.5 |
| HiP-256 | A+V       | 524,288 | 43.1 |
| Perceiver | A         | 480     | 38.3 |
| HiP-256 | A         | 122,880 | 41.3 |

4.3. PASCAL VOC

We also explore HiP for semantic segmentation, a dense visual task that aims to identify the object classes for each image pixel. Following prior literature (Grill et al., 2020; He et al., 2020), we pre-train both HiP and ResNet-50 on ImageNet. The pre-training procedure (masked auto-encoding followed by supervised fine-tuning) described in Section 3 is used to pre-train HiP. We use the PASCAL VOC 2012 dataset, fine-tuning on the commonly used train_aug2012 partition and report the mean intersection over union (mIoU) on the val2012 set.

To perform segmentation with HiP, we query the decoder using the fully learnt position embeddings to determine the class of each pixel. We either use the proposed hierarchical decoder (‘full’), or add a single-layer cross-attention decoder onto the bottleneck latents in HiP (cutting the HiP architecture in half, discarding the original decoder layers), similar to what we do for classification. Our ResNet-50 baseline uses the same fully convolutional segmentation head as used in (Grill et al., 2020; He et al., 2020). While the original Perceiver IO did not demonstrate results on segmentation (Jaegle et al., 2022), it can be easily applied for this task. The best ImageNet from-pixels Perceiver IO model (Jaegle et al., 2022) uses 8 repeated blocks, each with 6 self-attends (parameters shared between blocks). We also pre-train a leaner version of Perceiver IO (shallow) with better speed on ImageNet; this adapted model has just 1 block with one cross-attend and 12 self-attends. These Perceiver IO baselines use fixed Fourier position embeddings with 64
We note that the original PerceiverIO (denoted with a *), which makes it difficult for architectures with limited locality assumptions. We do not augment the data, nor do we employ specialized geometrical representations (these are quite important for driving performance in state-of-the-art methods, such as Xiang et al. 2021), so we compare only to our own baseline: a Perceiver IO model operating in the exact same experimental setting. We sweep over the number of self-attention layers (8, 12, 16), channels (512, 1024) and latents (256, 512, 1024) for this baseline, and for both models we sweep learning rate and weight decay — we report the best numbers across all sweeps. HiP-16 is used without modification.

Table 6 shows that the mIoU and training speed (in training steps/second, with fixed batch size) of three reference models and two HiP-16 variants. Sample segmentation outputs from HiP-16 are shown in Appendix Figure 8. Following common protocol, we use standard 512x512 resolution input, and train each model across 8 TPU-v3 cores. Both versions of HiP obtain mIoU on par with our optimized, convolutional ResNet-50 baseline, but are slightly slower than the ResNet-50. With its fewer layers, HiP (bottleneck) runs slightly faster than HiP (full), but this speed comes at the cost of 1 mIoU. Both HiP variants outperform our PerceiverIO baselines in both accuracy and training speed. We note that the original PerceiverIO (denoted with a *), may be underperforming (mIoU) because this very deep model overfits on the small PASCAL VOC dataset.

### 4.4. ModelNet40

As a last experiment we consider shape classification on ModelNet40 (Wu et al., 2015). Examples on this dataset are point clouds with 2048 points, each with 3 spatial coordinates (x, y, and z). The task is to classify each point cloud into one of 40 shape classes. The challenge with this type of data is that the points are randomly permuted, which makes it difficult for architectures with limited locality assumptions. We do not augment the data, nor do we employ specialized geometrical representations (these are quite important for driving performance in state-of-the-art methods, such as Xiang et al. 2021), so we compare only to our own baseline: a Perceiver IO model operating in the exact same experimental setting. We sweep over the number of self-attention layers (8, 12, 16), channels (512, 1024) and latents (256, 512, 1024) for this baseline, and for both models we sweep learning rate and weight decay — we report the best numbers across all sweeps. HiP-16 is used without modification.

Table 7 shows that HiP gets very slightly worse performance than Perceiver IO when training from scratch, possibly because the permutation invariance of Perceiver IO gives it a small edge. However, after MAE pre-training HiP-16 slightly outperforms Perceiver IO. The reason for this is not completely clear. We inspect the positional embeddings before and after MAE pretraining and they looked similar:

| Model                     | mIoU | Steps/Sec |
|---------------------------|------|-----------|
| ResNet-50 (He et al., 2016) | 70.5 | 14.8      |
| Perceiver IO (Jaegle et al., 2022)* | 62.0 | 7.2       |
| Perceiver IO (shallow)    | 67.7 | 9.9       |
| HiP-16 (bottleneck)      | 70.0 | 12.7      |
| HiP-16 (full)            | 71.0 | 11.4      |

Table 7. Results on ModelNet40 classification (top-1 accuracy, higher is better).

The main results of this paper is that positional embeddings cannot be learned for high-resolution signals using global classification losses alone, which is not the case for low-resolution signals. This observation may create more questions than answers: does it occur because these specific loss signals are too sparse for these highly overparameterized networks? Are the usual parameterizations of position features inadequate for high-resolution signals? or is classification simply poorly suited for learning high-resolution position features in general? We showed that dense masked auto-encoding is sufficient to learn good positional embeddings: could other approaches like regular auto-encoding, or (dense) contrastive learning approaches lead to results of similar quality? We leave these questions for future work.

### 5. Discussion

#### Connectivity learning. Perceiver and Transformer models can in principle learn any connectivity pattern in a soft way as they perform global attention. However, dense connectivity hinders scaling these models to a very large number of raw inputs. Convolutional networks employ extremely (hard) local operations and are hence very scalable with the number of inputs / outputs, but their local neighborhoods are hand-designed and must be tailored for different modalities or combinations thereof. The proposed HiP model bridges both families of models, enforcing a very loose, modality-agnostic local connectivity (that hinges on the assumption that the data exhibits explicit locality in the way it is presented, e.g. pixels not being randomly permuted). HiP still offers significant room for soft-connectivity learning within its encoder and decoder modules as it creates few local neighborhoods (groups). Most of its processing is still fully global and happens at HiP’s bottleneck.

The need for self-supervised pre-training. One of the main results of this paper is that positional embeddings cannot be learned for high-resolution signals using global classification losses alone, which is not the case for low-resolution signals. This observation may create more questions than answers: does it occur because these specific loss signals are too sparse for these highly overparameterized networks? Are the usual parameterizations of position features inadequate for high-resolution signals? or is classification simply poorly suited for learning high-resolution position features in general? We showed that dense masked auto-encoding is sufficient to learn good positional embeddings: could other approaches like regular auto-encoding, or (dense) contrastive learning approaches lead to results of similar quality? We leave these questions for future work.

Many small inputs. We are interested in building models that can take in arbitrary modalities. Some modalities will be low-dimensional but high-resolution, like those considered in this paper. There are however other modalities that may be high-dimensional but low-resolution. The proposed model will project all of them to latents of the same dimen-
Hierarchical Perceiver

sionality. The view we take here is that it may be preferable to operate on low-dimensional inputs, and break up high-dimensional ones into multiple smaller vectors, glued together by appropriate per-example learned embeddings. A more flexible solution may lead to better results, but we leave this for future work.

Augmentation: the last bastion of domain knowledge?
With HiP we demonstrate competitive results with the same model on various modalities with minimal preprocessing and prior knowledge – nowhere did we employ patching, convolutions, spectrograms, etc., even for very large inputs – only raw data. This is the setting that may be most promising for AutoML or anyomodal learning. However, we still exploit plenty of domain knowledge in the set of augmentations employed for each experiment. We believe this is only necessary because the datasets we experimented with are insufficiently large for anyomodal models, and to be competitive all methods currently rely on data augmentations. We suspect it will be possible to push anyomodal models even further without augmentation on larger datasets by developing better pretraining procedures.

Acknowledgements
We thank Andy Brock and Carl Doersch for helpful feedback.

References
Akbari, H., Yuan, L., Qian, R., Chuang, W.-H., Chang, S.-F., Cui, Y., and Gong, B. VATT: Transformers for multimodal self-supervised learning from raw video, audio and text. In Proceedings of Neural Information Processing Systems (NeurIPS), 2021.

Bao, H., Dong, L., and Wei, F. BEiT: BERT pre-training of image Transformers. In Proceedings of International Conference on Learning Representations (ICLR), 2022.

Brock, A., De, S., Smith, S. L., and Simonyan, K. High-performance large-scale image recognition without normalization. In Proceedings of International Conference on Machine Learning (ICML), 2021.

Dai, Z., Lai, G., Yang, Y., and Le, Q. V. Funnel-Transformer: Filtering out sequential redundancy for efficient language processing. In Proceedings of Neural Information Processing Systems (NeurIPS), 2020.

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

Fan, H., Xiong, B., Mangalam, K., Li, Y., Yan, Z., Malik, J., and Feichtenhofer, C. Multiscale vision Transformers. In Proceedings of IEEE International Conference on Computer Vision (ICCV), 2021.

Fischer, P., Dosovitskiy, A., Ilg, E., Häusser, P., Hazırbaş, C., Golkov, V., van der Smagt, P., Cremers, D., and Brox, T. FlowNet: Learning optical flow with convolutional networks. In Proceedings of IEEE International Conference on Computer Vision (ICCV), 2015.

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.

Grill, J.-B., Strub, F., Altché, F., Tallec, C., Richemond, P. H., Buchatskaya, E., Doersch, C., Pires, B. A., Guo, Z. D., Azar, M. G., et al. Bootstrap your own latent: A new approach to self-supervised learning. In Proceedings of Neural Information Processing Systems (NeurIPS), 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.

He, K., Fan, H., Wu, Y., Xie, S., and Girshick, R. Momentum contrast for unsupervised visual representation learning. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2020.

He, K., Chen, X., Xie, S., Li, Y., Dollár, P., and Girshick, R. Masked autoencoders are scalable vision learners. arXiv preprint arXiv:2111.06377, 2021.

Hsu, W.-N., Bolte, B., Tsai, Y.-H. H., Lakhota, K., Salakhutdinov, R., and Mohamed, A. HuBERT: Self-supervised speech representation learning by masked pre-
Hierarchical Perceiver

diction of hidden units. arXiv preprint arXiv:2106.07447, 2021.

Huang, G., Sun, Y., Liu, Z., Sedra, D., and Weinberger, K. Q. Deep networks with stochastic depth. In Proceedings of European Conference on Computer Vision (ECCV), 2016.

Jaegle, A., Gimeno, F., Brock, A., Zisserman, A., Vinyals, O., and Carreira, J. Perceiver: General perception with iterative attention. In Proceedings of International Conference on Machine Learning (ICML), 2021.

Jaegle, A., Borgeaud, S., Alayrac, J.-B., Doersch, C., Ionescu, C., Ding, D., Koppula, S., Zoran, D., Brock, A., Shelhamer, E., Hänaiff, O., Botvinick, M., Zisserman, A., Vinyals, O., and Carreira, J. Perceiver IO: A general architecture for structured inputs & outputs. In Proceedings of International Conference on Learning Representations (ICLR), 2022.

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.

Likhosherstov, V., Arnab, A., Choromanski, K., Lucic, M., Tay, Y., Weller, A., and Dehghani, M. PolyViT: Co-training vision Transformers on images, videos and audio. arXiv preprint arXiv:2111.12993, 2021.

Liu, Y., Ott, M., Goyal, N., Du, J., Joshi, M., Chen, D., Levy, O., Lewis, M., Zettlemoyer, L., and Stoyanov, V. RoBERTa: A robustly optimized BERT pretraining approach. arXiv preprint arXiv:1907.11692, 2019.

Liu, Z., Lin, Y., Cao, Y., Hu, H., Wei, Y., Zhang, Z., Lin, S., and Guo, B. Swin Transformer: Hierarchical vision Transformer using shifted windows. In Proceedings of IEEE International Conference on Computer Vision (ICCV), 2021.

Loshchilov, I. and Hutter, F. Decoupled weight decay regularization. In Proceedings of International Conference on Learning Representations (ICLR), 2019.

Nagrani, A., Yang, S., Arnab, A., Jansen, A., Schmid, C., and Sun, C. Attention bottlenecks for multimodal fusion. In Proceedings of Neural Information Processing Systems (NeurIPS), 2021.

Nawrot, P., Twokowski, S., Tyrolski, M., Kaiser, Ł., Wu, Y., Szegedy, C., and Michalewski, H. Hierarchical Transformers are more efficient language models. arXiv preprint arXiv:2110.13711, 2021.

Newell, A., Yang, K., and Deng, J. Stacked hourglass networks for human pose estimation. In Proceedings of European Conference on Computer Vision (ECCV), 2016.

Ranftl, R., Bochkovskiy, A., and Koltun, V. Vision Transformers for dense prediction. In Proceedings of IEEE International Conference on Computer Vision (ICCV), 2021.

Ronneberger, O., Fischer, P., and Brox, T. U-Net: Convolutional networks for biomedical image segmentation. In International Conference on Medical Image Computing and Computer Assisted Intervention (MICCAI), 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.

Vincent, P., Larochelle, H., Bengio, Y., and Manzagol, P.-A. Extracting and composing robust features with denoising autoencoders. In Proceedings of International Conference on Machine Learning (ICML), 2008.

Wang, J., Sun, K., Cheng, T., Jiang, B., Deng, C., Zhao, Y., Liu, D., Mu, Y., Tan, M., Wang, X., et al. Deep high-resolution representation learning for visual recognition. IEEE Transactions on Pattern Analysis and Machine Intelligence, 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.

Xiang, T., Zhang, C., Song, Y., Yu, J., and Cai, W. Walk in the cloud: Learning curves for point clouds shape analysis. In Proceedings of IEEE International Conference on Computer Vision (ICCV), 2021.

Yan, S., Yang, Z., Li, H., Guan, L., Kang, H., Hua, G., and Huang, Q. Implicit autoencoder for point cloud self-supervised representation learning. arXiv preprint arXiv:2201.00785, 2022.
Hierarchical Perceiver

Figure 5. Visualizing the inner product distance between learned positional embeddings within and across groups for a HiP-16 model trained on ImageNet. Each image shows the inner product distance between a single pixel (taken 14 pixels apart, across the image, marked in black arrows for the first row) and all other pixels. As can be seen learned embeddings are close for nearby pixels within each group, as well as for similar locations across different groups (up to phase). Note that each group in HiP-16 corresponds to a set of 14 consecutive pixel rows.

A. Appendix

A.1. Additional positional embedding visualization

We include more analysis of the learned HiP-16 positional embeddings in this section.

Figure 5 shows distances between the positional embeddings of different pixels. Generally, the learned embeddings seem to mostly serve the purpose of relating pixels within each group; this may be an effective strategy for obtaining compressed position codes as it obviates the need to avoid global collisions – codes can repeat in other groups. It is possible that for establishing more global locations within the image the model uses instead the learned latent vectors associated with each group.

Figure 6 shows a factorization of the 32-d positional embeddings into their top principal components, which exhibit connections to Fourier basis.

A.2. Masked auto-encoding validation loss curves

We show example curves for validation losses on ImageNet in Figure 7, using 85% masking. Note that a version of HiP using Fourier positional embeddings gets considerably lower loss which suggests that they enable more precise pixel lookup than the low-dimensional learned ones. Results in the paper suggest that this level of precision may not be necessary for recognition and labeling tasks.
A.3. Relationship to Swin and ViT-based models

ViT (Dosovitskiy et al., 2021) and more generally standard Transformers (Vaswani et al., 2017) scale the amount of feature extraction they do by the input size – e.g., if a ViT model processes 1024 patches, it will compute many more features than if it processes 256 patches. The bulk of computation in Perceivers occurs at layers whose size is independent of the input size, as they project to and operate in a fixed-size latent space. This same principle makes it possible to do global computation efficiently in the single-group processing bottleneck of HiP, independently of input resolution, unlike Swin (Liu et al., 2021), where computation scales with input resolution. To deal with this Swin keeps computation local throughout the model, like ConvNets, and requires different “groups” within the network to communicate regularly, which is implemented via an iterated shifting operation. This is elegant but is not especially friendly for some types of hardware, like TPUs. We took inspiration from Swin for some aspects of our model’s design, for example in choosing the pattern of number of self-attention layers and self-attention heads.

A.4. Visualizing Sample Segmentations on PASCAL VOC

Provided in Figure 8 are visualizations for the segmentation outputs of a HiP-16 model on six randomly sampled images from the PASCAL VOC val2012 set.
Figure 6. Channel images of top PCA-projected spatial embeddings for HiP-16 trained on ImageNet, sorted left-to-right, top-to-bottom. As can be seen within each group (sets of 14 consecutive pixel rows), learned features resemble specific Fourier basis features.
Figure 7. Validation HiP-16 masked auto-encoding losses (mean-squared pixel error) using 85% masking on ImageNet 224x224 images with either 32-d Learned or 256-d Fourier positional embeddings.
Figure 8. Sample segmentation outputs of HiP-16 on PASCAL VOC 2012 images. Left column: the original image. Middle left column: the ground truth segmentation. Middle right column: class-colored, HiP-16 segmentation prediction of the image. Right column: the HiP-16 segmentation prediction overlaid on top of the original image. Note that the outputs are quite sharp because the model operates directly on pixels.