

Zorro: the masked multimodal transformer

Adrià Recasens^{1†} Jason Lin¹ João Carreira¹ Drew Jaegle¹ Luyu Wang¹ Jean-baptiste Alayrac¹
Pauline Luc¹ Antoine Miech¹ Lucas Smaira¹ Ross Hemsley¹ Andrew Zisserman^{1,2}

¹DeepMind ²VGG, Dept. of Engineering Science, University of Oxford

Abstract

Attention-based models are appealing for multimodal processing because inputs from multiple modalities can be concatenated and fed to a single backbone network – thus requiring very little fusion engineering. The resulting representations are however fully entangled throughout the network, which may not always be desirable: in learning, contrastive audio-visual self-supervised learning requires independent audio and visual features to operate, otherwise learning collapses; in inference, evaluation of audio-visual models should be possible on benchmarks having just audio or just video. In this paper, we introduce Zorro, a technique that uses masks to control how inputs from each modality are routed inside Transformers, keeping some parts of the representation modality-pure. We apply this technique to three popular transformer-based architectures (ViT, Swin and HiP) and show that with contrastive pre-training Zorro achieves state-of-the-art results on most relevant benchmarks for multimodal tasks (AudioSet and VGGSound). Furthermore, the resulting models are able to perform unimodal inference on both video and audio benchmarks such as Kinetics-400 or ESC-50.

1. Introduction

Our perception of the world is inherently multimodal: humans and other animals effortlessly integrate many modalities to build their view of the world [5, 23]. Although multimodal integration can help construct a richer perspective on reality [10, 44], humans can easily process information and perform tasks even when only a single modality (e.g. sound, vision, or touch) is present [11, 32, 45]. However, this flexibility is hard to find in perceptual computational models. Architectures for multimodal perception have typically been divided on early fusion, mid-fusion and late-fusion, but most of them need all modalities to be present in order to operate. With human flexibility as an inspiration, in this paper we introduce *Zorro*, a multimodal Transformer archi-

ture which is able to operate in both a single-modality and multi-modality setting. This property improves the overall performance of the model while opening the door to off-the-shelf self-supervised pre-training.

Our key architectural innovation in Zorro is to create separate unimodal and multimodal (fusion) representation streams within a single standard Transformer backbone. We achieve this without engineering the architecture, but instead by applying appropriate masks in all attention operations, resulting in some outputs that only capture individual modalities and some outputs that capture multimodal information. This has the direct benefit that the model can be applied when a subset of modalities is absent, e.g. a model trained on audio and video can be evaluated on audio alone.

While most of the emphasis of novel developments in the supervised space is put on the architecture, the unimodal outputs can be further exploited by introducing additional self-supervised training schemes. In contrast to recent multimodal attention-based models [27, 36] that entangle both modalities throughout the network, Zorro supports self-supervised contrastive training in a single network without representation collapse, thanks to its unimodal outputs (see Figure 1). In this work, we explore this possibility by pre-training our model with an audio-visual contrastive loss [3]. Differently from previous work, we can do this pre-training without the necessity of separate backbones per modality.

This paper presents four contributions: **(a)** we introduce Zorro, a novel set of Transformer-based multimodal architectures which enable both supervised and self-supervised training and, once trained, can be used for multimodal or unimodal inputs; **(b)** we introduce three Zorro-based architectures using state-of-the-art models such as ViT, SWIN and HiP; **(c)** we show that Zorro can be pre-trained on a large-scale audio-visual dataset in a self-supervised manner, and can also be pre-trained on unimodal datasets; and **(d)** we benchmark our resulting models on AudioSet, VGGSounds, Kinetics-400 and ESC-50. The model achieves state-of-the-art performance when compared with previous self-supervised learning techniques on most relevant benchmarks, while also achieving comparable performance with previous work for supervised training with labels.

[†]Correspondence to: Adrià Recasens (arecasens@google.com)

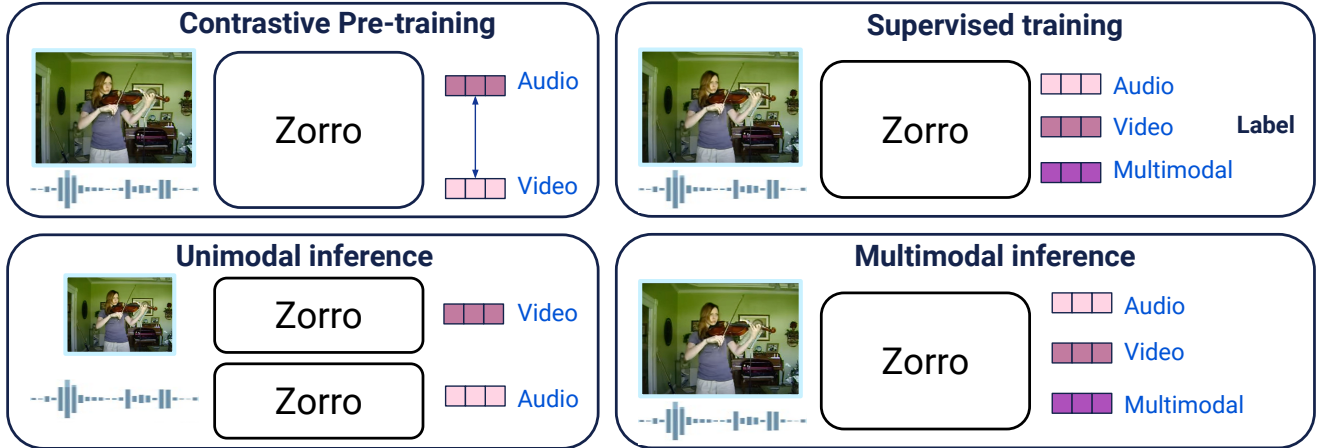


Figure 1. In this paper we introduce the Zorro multimodal architecture which enables both self-supervised contrastive learning and supervised learning. When used for self-supervision, the single-modality outputs are used together with a standard cross-modal self-supervised loss. When used for supervised learning, all outputs can be used for the final classification.

2. Related Work

Multimodal perception: Multimodal perception is challenging as data from the various modalities can have different topologies, temporal frequencies and relative importances that depend on each task [9]. With the emergence of convolutional neural networks, numerous works fused activations from intermediate tensors [7, 14, 19, 21, 47, 52, 53], but this required considerable engineering, as different modalities come in differently shaped feature grids and there are many different ways to combine them.

Self-supervised audio-visual learning: Various methods have been used to employ the cross-modality similarity as a self-supervisory signal [4, 6, 7, 30, 35, 37, 39, 43]. Most approaches rely on single-modality backbones which produce representations which are used in the self-supervised loss [3, 4, 39, 41]. These techniques process different modalities with different sets of weights and restrict the ability to reason across modalities. Less common are approaches which learn self-supervised models with multiple modalities at once. One recent work in this direction is [46], which learns representations using audio, video and text. However, to avoid the collapse of the self-supervised loss, they feed the modalities two at a time, increasing the amount of necessary forward passes. Instead, Zorro masking can produce unimodal outputs without running the model multiple times.

Transformer architectures: Inspired by ViT [18], follow up work proposed single-modality processing for video [8] and audio [24] using patch-based encodings. Transformer-based methods have also been proposed to tackle audio-visual classification. The closest to our method is MBT [36], which builds a multimodal architecture out of single-modality Transformers for video [8, 18] and audio [24]. MBT merges modalities by creating an attention bottleneck

which restricts communication between the audio and visual heads. Our method also regulates cross-modality communication, but by masking the latent connections we are able to obtain modality-specific heads while in MBT the representation is entirely multimodal. Another relevant work is VATT [2], a Transformer-based architecture to model video, audio and text with a single backbone. Differently from our work, in VATT each modality is independently processed by the transformer. Finally, the Perceiver architecture [27] scales to a large number of inputs by cross-attending to a set of latent queries. In this work, we use the follow-up Hierarchical Perceiver [13] which splits inputs and outputs into groups to improve model efficiency.

Masking attention in Transformers: The original transformer architecture [50] used attention-masking for language modelling. After the success of image-based architectures, alternatives have been proposed to use attention masking to alleviate computational requirements of the architecture. Swin [34] proposed the use of local windows, restricting the self-attention layers to only neighbour pixels. Furthermore, mask2former [16], also restricted the cross-attention to local regions, enabling the use of transformers for high dimensional output (e.g segmentation).

3. Zorro: the masked multimodal Transformer

In this paper, we introduce Zorro, a multimodal architecture which enables both supervised and self-supervised training. In this section, we unpack how Zorro accomplishes this using modality-aware masking and by repurposing the original transformers components to allow contrastive learning between modalities. The key innovation of the architecture is introducing separate latent allocations for the different modalities, leading to a final representation which is par-

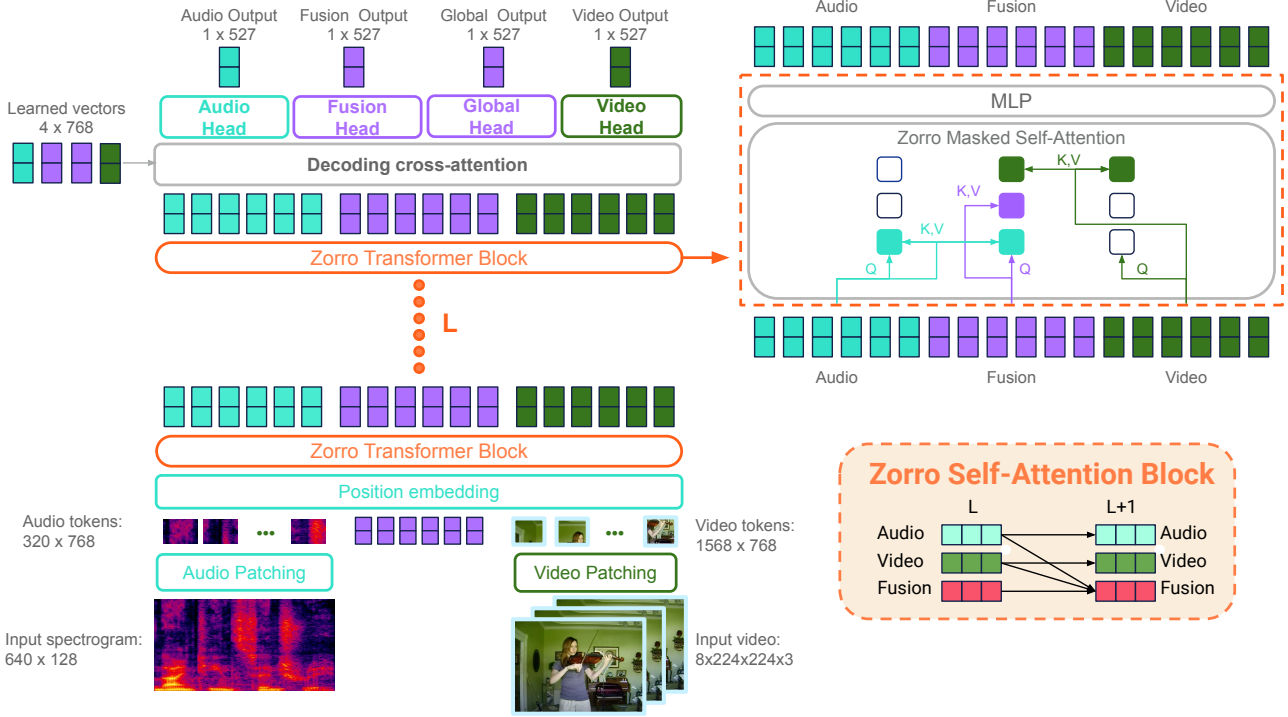


Figure 2. **The Zorro-ViT model architecture:** The input to our model are video frames and audio spectrograms. Each of those inputs is patched using a 2D convolution and projection to input dimension D . Both audio and video input tokens are concatenated with a set of learned fusion vectors and added a position embedding. Next, we process these inputs through L Zorro’s self-attention layers, where the Zorro masking is applied. Specifically, our masking strategy blocks the information to flow towards the unimodal hidden representation, while still allowing the fusion representation to access all modalities. By doing this, we ensure that the image and audio representations are gated access to (i.e. depend on) only the video and audio inputs respectively. To produce the outputs, we learn a set of queries that cross-attend (also using masked attention) to the unimodal and multi-modal representation.

tially unimodal (part of the representation sees only a single modality) and partially multimodal (part of the representation can attend to all modalities). First, we will describe Zorro applied to the ViT architecture. Second, we extend Zorro to two other state-of-the-art transformer architectures, Swin and HiP. Finally, we end this section by describing how to use Zorro for self-supervised contrastive learning.

3.1. Architecture

Zorro-ViT overview. Figure 2 depicts the Zorro architecture, which consist of three main blocks. First, Zorro processes the data in form of patches (similar to ViT [18]). In this stage, data from each modality is first converted into a 2D array of representations. This can be done by either (i) dividing the input tensor into sequential groups (either points or patches) and applying a linear projection, or (ii) applying domain-specific processing such as 1D/2D/3D convolutions and flattening. We use a 2D convolution to extract 16×16 patches and project them to the input dimension D . Next, position embeddings are added to the projected vectors so that the model is able to localise and distinguish

each embedded patch. Learned multimodal fusion vectors are then introduced. Second, the resulting tokens are concatenated to form a single set and are then processed by L layers of a Transformer [50] with Zorro masking. Finally, to produce the final output we learn a set of queries that cross-attend to the output of the last self-attention layer similar to PerceiverIO [26]. We utilise the standard cross-attention operation [26], and produce 4 different outputs: an audio output, a video output, a fusion output (which only sees the multi-modal part of the representation) and a global output that sees the whole representation. These three steps are described in more detail next.

Input pre-processing. Let $x = (x_v, x_a)$ be a video sample consisting of frames $x_v \in \mathbb{R}^{N_f \times H \times W \times 3}$ and audio spectrogram $x_a \in \mathbb{R}^{T \times N_s}$ where N_f is the number of frames, N_s the dimensionality of the spectrogram, H is the height of the frame, W is the width of the frame and T is the number of temporal steps in the spectrogram. To downscale the input, we use a 2D convolution f^{patch} which yields $u = (u_v, u_a) = (f_v^{\text{pre}}(x_v), f_a^{\text{pre}}(x_a))$. Arrays (u_v, u_a) are

then flattened and absolute learned position encoding are added. Finally, we learn a set of n_{fusion} latent vectors which are concatenated to the audio and video input tokens.

Masked attention. The key contribution of this paper is splitting the Transformer representation into specialised groups. Using masked attention we force part of the representation to attend only to itself, while other parts can attend to the whole representation. The main goal of this approach is to split the representation in three parts: a part which only focuses on video tokens, a part which focuses on audio tokens, and the remaining vectors which can attend to the whole representation.

We mask two parts of the model: the self-attention [50] and the decoding cross-attention [26]. Both parts consist of the same underlying operation which takes keys k , values v and queries q to produce the final output o . To this end, we introduce a masking binary tensor m that specifies which vectors are connected to each other. Entries of the masking matrix are $m_{ij} = 1$ if information can flow from latent i to latent j . By setting $m_{ij} = 0$, we indicate to the model that this connection should be omitted. This mask is applied to the standard attention output operation $o_i = \sum_j a_{ij} \cdot v_j$ which becomes $o_i = \sum_j \hat{a}_{ij} \cdot v_j$ where:

$$\hat{a}_{ij} = \frac{m_{ij} \exp(\frac{q_i^\top k_j}{\sqrt{D}})}{\sum_{\{j', m_{ij'}=1\}} \exp(\frac{q_i^\top k_{j'}}{\sqrt{D}})}. \quad (1)$$

In contrast to MBT [36], our modality-specific representation does not have access to the global representation, which prevents cross-modality information flows. Specifically, we set $m_{ij} = 1$ if j is a part of the fusion representation, otherwise we only set $m_{ij} = 1$ if i and j are vectors of the same modality. By doing this, we explicitly prevent information from the fusion stream leaking into the unimodal representation. This is the key to preserving pure streams that correspond to single modalities.

Output space. In ViT architecture, a learnable CLS token is used to produce the output embedding vector. Instead, inspired by the PerceiverIO [26], we learn a set of decoding vectors which are used to query the output from the Transformer to produce the final output. Each decoding vector cross attends to a subset of tokens to produce the final output vector. This decoding strategy can be used to produce as many outputs as desired, opening up the possibility for dense tasks such as segmentation or flow estimation.

As we are relying on having the Transformer representation split into specialised groups, we need to also apply Zorro’s masking to the output cross attention. Specifically, we found it beneficial to define four outputs for our model. The audio-specific output o_A , which only contains information coming from the audio input. The video-specific output

o_v , which only includes information from the video modality. The fusion specific output o_F , which is computed by attending only to the fusion stream. And finally, a global output o_G , which attends to all the outputs in the model. Although o_G and o_F do contain similar information, we found it useful to still keep two different heads.

3.2. Extending Zorro for other architectures

In this section, we propose variants of Zorro for two state-of-the-art attention-based architectures, Swin and HiP. Differently from the ViT implementation, when building Zorro-Swin and Zorro-HiP we use the specific architecture building block for each modality and the fusion stream while we join the modalities with a cross-attention operation. This is required as the ViT masking is not directly applicable to Swin and HiP, but the overall idea remains the same.

Zorro-Swin: Swin [34] is a ViT-inspired transformer architecture which has shown improved efficiency and performance. The main innovation versus the original ViT architecture is to apply the self-attention operations on nearby tokens instead of all tokens in the input image. This reduces computational requirement while allowing the model to perform bottom-up inference. In order to build Zorro-Swin, our main modification to the original architecture is to process individual modalities using Swin transformers. At the end of each Swin block, we update the fusion representation by cross-attending to both the unimodal and multimodal representation. To process the fusion representation, we use the same self-attention as in Zorro-ViT. Given this design, we are free to use different architectures to process each modality. We use the original 2D Swin [34] to process the audio spectrograms while our adaptation of the Swin architecture for video. Similarly to Zorro-ViT, no multimodal information flows into the unimodal streams. Detailed description of Zorro-Swin can be found in Section A in the Appendix.

Zorro-HiP: The hierarchical perceiver [13] extends the previously introduced Perceiver models [26, 27] models, by splitting the inputs into groups, and operating only within those groups. Through the hierarchical architecture, those groups fuse together in order to aggregate information and globally reason about the input. In our implementation of HiP, instead of using directly the pixels and audio signal as input, we create patches similarly to the ViT/Swin implementation. In order to create Zorro-HiP, we use HiP building blocks for each modality. Specifically, those blocks group the inputs into smaller sets, cross-attend using learned features and finally apply self-attention layers to the outputs of the cross attention operation (see [13] for more details). In order to update the fusion representation, we learn a set of queries which cross attend to both unimodal and multimodal representation per each layer. More details can be found in Section A in the Appendix.

3.3. Contrastive learning with Zorro

Contrastive audio-visual methods learn representations by aligning audio and video into a common embedding space. As opposed to unimodal approaches, instead of producing multiple views of the data, they use different modalities as views. One important requirement is for the two backbones to not share information. If information is shared across modalities, the self-supervised training can easily collapse or converge to a trivial solution.

Models for multimodal perception typically produce a single output for the multiple inputs. This is sufficient for supervised applications, but prevents the use of these audio-visual contrastive techniques. We design Zorro in order to process unimodal and multimodal outputs, with the intention of enabling the use of self-supervised contrastive losses.

Noise Contrastive Estimation: For training with the standard noise-contrastive estimation loss, we follow the implementation of the audio-visual loss from [3]. Given the audio output o_a and the video output o_v , we apply a final linear projection (different per modality) g_a and g_v to yield the final embedding vectors: $z_a = g_a(o_a)$ and $z_v = g_v(o_v)$. We compute the similarity between z_a and z_v by taking a normalised dot product and dividing by a temperature parameter τ , $\text{sim}(z_a, z_v) = \exp(\frac{z_a z_v}{\tau})$. Finally we apply the NCE loss:

$$L_{\text{NCE}}(z_a, z_v) = - \sum_i \log \frac{\text{sim}(z_a^i, z_v^i)}{\sum_{j,k} \text{sim}(z_a^k, z_v^j)} \quad (2)$$

Equation 2 introduces describes the loss for audio-visual contrastive training. However, this technique does not train any parameters specific to the fusion representation or output (e.g. the fusion cross-attention or the fusion weights if the model has separate weights per modality). In order to self-supervise the output of the fusion stream, we add a fusion-visual and fusion-audio contrastive loss. We define a self-supervised loss contrasting both unimodal representations (audio and video) separately with the multimodal one (fusion). With those changes, the new loss is:

$$L_{\text{NCE}} = L_{\text{NCE}}(z_a, z_v) + L_{\text{NCE}}(z_a, z_f) + L_{\text{NCE}}(z_I, z_f) \quad (3)$$

4. Experiments

In this section, we evaluate the Zorro architecture on multiple settings. We first present details of the training and evaluation procedures, as well as the main datasets we use. We evaluate the method against state-of-the-art models on three standard audiovisual benchmarks (AudioSet [22], VGGSound [15] and Kinetics-400 [14]), one vision benchmarks (Kinetics-400 [14]) and one audio benchmark (ESC-50 [40]). Finally, we ablate the main design decisions that drove our research and showcase Zorro’s flexibility. Specifically, we

compare the different architectures, study the effect of missing modalities, pre-train Zorro with unimodal data and explore alternative attention-masking strategies.

4.1. Experimental details

In order to showcase Zorro’s ability to reason across different modalities, we pre-train it using self-supervision as well as with standard supervision using class labels. In this section, we provide the most important details of the training procedure. Additional details about inputs, architectures and training can be found in Section A and B in the Appendix.

Pre-training datasets: We utilise four datasets for pre-training: AudioSet [22], YouTube-8M, ACAV-100M [33] and ImageNet-21k [42]. AudioSet consist of 1.9M videos which contain 527 classes of annotated sounds. As the dataset is highly unbalanced, [36] proposed a smaller more balanced variant of the training set with 500k examples. For the ablation experiments and training from scratch, we use the 1.9M version while for fine-tuning we also use AudioSet-500k for fair comparison with the state-of-the-art. YouTube-8M [1] consist of 8M videos with audio and visual frames, annotated in a multi-label fashion with 3862 different classes. Videos are representative of many activities, resulting a very natural distribution of data. ACAV-100M consist of 100M videos with audio and visual frames without associated labels, which have been curated to contain a strong audio-visual correlation. We use 59M of those videos for self-supervised learning. ImageNet-21k consist of 13M images annotated on 21k classes, and been typically used for large-scale pretraining of visual transformer models [18].

Audio-visual evaluation benchmarks: To evaluate the ability of Zorro to learn and transfer multimodal representations, we evaluate on standard audio-visual benchmarks. Specifically, we evaluate Zorro in AudioSet, VGGSound [15] and Kinetics-400 [28]. VGGSound consists of 163, 603 training and 13579 test samples drawn from 10-second YouTube videos which span 309 single-label, mutually exclusive classes. It focuses on real life audio evaluation with audio-visual correspondence where sounds are visually evident in the video. Kinetics-400 consists of 201K training videos of everyday actions which are classified into 400 unique classes. While some datasets have bias in audio or video modality, Zorro is able to learn the extent to rely on each modality.

Unimodal evaluation benchmarks: Zorro can be trained on multi-modal data but evaluated on unimodal data. To further show this we evaluate the multi-modal trained Zorro models on unimodal fine-tuning tasks: Kinetics-400 for vision and ESC-50 for audio. ESC-50 dataset contains 2k clips classified into 50 unique classes.

Zorro inputs: The inputs to our model are video and audio. The audio and video are synced and cover the same

Table 1. **AudioSet-2M comparison: training from scratch.** We report the performance of our models trained on audio-visual data compared with the state-of-the-art when trained from scratch. We report the mean average precision on the AudioSet test set.

Model	Train Mod	Eval Mod	AudioSet
HiP [13]	A+V	A+V	43.8
Perceiver [27]	A+V	A+V	44.2
ERANN [51]	A	A	45.0
Zorro-ViT	A+V	A+V	45.1
Zorro-HiP	A+V	A+V	45.2
Zorro-Swin	A+V	A+V	46.5

time span. Video consists of 8 frames of size 224×224 . When training in AudioSet, we sample videos at 3.12FPS which results on 2.56s of audio and video. Specific FPS per model and audio length for pre-training and fine-tuning is reported in Section B in the Appendix. During training, we use random cropping as well as color augmentation in frames. For ESC-50, we match the lengths of the pre-trained model, looping over the audio sequence if required. Audio is sampled at $48k Hz$, converted to spectrograms as inputs to our model using 128 bins. To augment the audio in training, we use SpecAugment [38] and frequency jittering. During evaluation, we subsample the input video and audio into multiple equally spaced clips and average their predictions.

Architectural details: Zorro is based on unimodal transformer architectures (ViT, Swin and HiP), adapted for multi-modal processing (similar to [36]). Through all our experiments we use ViT-B/16. For details on ViT, Swin and HiP architecture, see Section A in the Appendix.

Training details: We use the Adam optimiser with cosine decay learning rate schedule, weight decay and learning rate warmup. When fine-tuning, for Zorro-ViT and Zorro-Swin we find better to use SGD optimiser and momentum 0.9. We train all models for 50 epochs except for the ACAV-100M datasets where we train for 10 epochs and the *input-level* and *bottleneck* baselines where we train for 25 to prevent severe overfitting. We find best to use $n_{\text{fusion}} = 6$ in all models. For AudioSet fine-tuning, we use mixup ($\alpha = 0.3$) and label smoothing. We use cross-entropy loss for uni-label datasets and binary sigmoid cross-entropy for multi-label. We train one classifier for each of the 4 outputs of the model and average its predictions. For contrastive training, we follow the procedure outlined in Section 3.3.

4.2. State-of-the-art comparison

Next, we evaluate Zorro against state-of-the-art methods. We evaluate our audio-visual trained Zorro on benchmarks for audio-visual classification, video classification and audio classification, showcasing the universality of the approach.

Training AudioSet-2M from scratch: First, we evaluate Zorro when trained from scratch on AudioSet-2M using both the audio and visual modalities. Table 1 reports that Zorro matches or overperforms other methods that directly trained on AudioSet-2M from scratch. Note that PlayItBack [48] is not listed in Table 1 as it was trained with AudioSet-500k. This setting shows the model’s ability to adapt to the multi-modal inputs without the need of pre-trained data.

Multi-modal comparison: We train and evaluate our pre-trained models on AudioSet-500k (see [36] for details), VGGSound and Kinetics-400 where we use both the audio and visual inputs. Similar to [36], for Zorro-ViT we allocate different weights for the audio, video and fusion latents. We found this useful for improving the fine-tuning accuracy. Table 2 reports the performance of our models. We divide the table into two different parts. First, we report the Zorro performance when contrastive self-supervision is used for pre-training (no labels). Zorro improves over all previous works on AudioSet and VGGSound. In AudioSet, our best-performing model on that setting is only 1.2% away from Zorro with supervised pre-training, which demonstrates the ability of the self-supervised pre-training technique for learning general features. In VGGSound, Zorro performs similarly with the supervised state-of-the-art when pre-trained only with self-supervision. Finally, for Kinetics-400, the resulting performance is not far from models with supervised pre-training. In the bottom part of the table we report the performance of Zorro when using supervised pre-training. We include the performance of the model when initialized with ViT pre-trained on ImageNet-21k. Even without multi-modal pre-training, Zorro is able to perform more than 1% better than existing SOTA models in AudioSet. When pre-trained on YouTube-8M, Zorro improves its performance as a result of the multi-modal nature of its pre-training. The final performance in AudioSet represents a 1.9% improvement over the state-of-the-art, MBT [36]. Furthermore, unlike Zorro, MBT cannot perform unimodal inference when trained with multi-modal data. Note, we have not demonstrated it here, but Zorro can also be trained using unimodal self-supervised methods such as MAE [25] and DINO [12] separately on the audio and visual streams. We discuss supervised unimodal training below.

Video comparison: To showcase Zorro’s performance in the unimodal regime, we fine-tune our models (pre-trained on audio and video) on the task of video classification for Kinetics-400 using only video. Table 2 reports the results. Our goal is not to show state-of-the-art performance on this setting, as we are aware of the improvements made on Transformer architectures to solve that task [34, 54, 55]. Our goal is to provide an efficient mechanism for pre-training those architectures in order to improve the final performance on unimodal and multimodal inference. When Zorro is pre-

Table 2. **State-of-the-art results:** We compare Zorro with the state-of-the-art in two settings: when labels are not used in pre-training or when labels are used. We report the mean average precision on the AudioSet test set and top-1 accuracy on K-400, VGGSound and ESC-50. IN-21k is ImageNet-21k [42], YT8M is YouTube-8M [1], ACAV is ACAV-100M [33] and K-400 is Kinetics-400 [14]. When using ImageNet-21k initialisation, we use the pre-trained weights to initialise the video, audio and fusion parameters.

Model	Pre-Training			Eval: Video+Audio			Eval: Video	Eval: Audio
	Dataset	Sup/SSL	Mod	AS	VGGSound	K-400	K-400	ESC-50
No pre-training								
SlowFast R101-NL [20]						79.8	79.8	
AVSlowFast [53], R101						78.8		
AudioSlowFast [29]					52.5			
ERANN [51], R101				45.0				89.2
PlayItBack [48], R101				47.7	53.7			
Self-supervised pre-training								
MaskSpec [17], ViT	AS	SSL	A	47.1				89.6
Zorro-HiP	ACAV	SSL	A+V	49.4	61.3	67.9	64.6	88.4
Zorro-Swin	ACAV	SSL	A+V	49.4	61.1	73.7	69.4	91.4
Zorro-ViT	ACAV	SSL	A+V	50.3	63.6	76.5	74.1	93.6
Supervised pre-training								
ViViT-Base [8]	IN-21k	Sup.	V			80.0	80.0	
MaskSpec [17], ViT	AS	Sup	A					98.2
PaSST [31]	IN	Sup.	V	49.6				96.8
AST [24]	IN-21k	Sup.	V	45.9				95.7
MBT [36], ViT	IN-21k	Sup.	V	49.6	64.1	80.8	79.4	
Zorro-ViT	IN-21k	Sup.	V	50.9	63.1	79.8	77.6	81.7
Zorro-ViT	YT8M	Sup.	A+V	51.5	64.8	79.6	76.1	93.1

trained using a contrastive loss and fine-tuned on Kinetics-400 (video only), Zorro-ViT performs only 2.4% worse than when using audio-visual input. This shows the robustness of our model when reduced to using a single modality. Furthermore, when using the Zorro model pre-trained on YT8M, our model is able to perform similarly to comparable architectures. Alternative to fine-tuning, we can also use the audio-visual trained model (column *Audio+Video*) and only feed the video. In that setting, our model trained on YouTube-8M performs at 76.3 top-1, on par with the video only fine-tuned result. This unimodal inference on a multi-modal trained model is not possible with MBT, where retraining is needed.

Audio comparison: To evaluate Zorro’s audio capabilities, we fine-tune our models on ESC-50 (audio-only dataset) and report results in Table 2. When pre-trained on YouTube-8M, Zorro performs close to AST, an specialised audio transformer comparable in size. When using self-supervised pre-training, Zorro improves performance over previous methods; Zorro-ViT has an accuracy of 93.6%, close to state-of-the-art supervised methods.

4.3. Architecture comparison

In this section, we discuss the different architectures introduced in this paper. In Table 3 we report comparison for

those architectures in two settings: when trained from scratch and when pre-trained with an audio-visual contrastive loss followed by a linear layer on top, using Audioset-2M. When training from scratch, we observe Zorro-Swin performs the best across the different models, both in the supervised and contrastive regimes. Although the number of parameters is larger than ViT, Swin trains 25% faster than ViT. HiP is the fastest of the three, while not losing much on accuracy. See Section A in the Appendix for model speed comparison. Furthermore, in Table 2 we also present the results of fine-tuning these architectures after contrastive pre-training. It is important to note that for ViT, in this table we use one set of parameters per modality, which significantly increases the parameter count (98M to 267M). In this regime, we observe how ViT is the best. However, Swin and HiP are faster and retain most of the performance.

4.4. Zorro model flexibility

Unimodal inference with a multimodal backbone: Here we study the ability of audio-visual trained Zorro to produce meaningful unimodal outputs when fed with unimodal data. To achieve this we zero out the missing modality and only provide useful inputs for one modality, either video or audio. Results are reported in Table 3. Models without unimodal

Table 3. **Masking configurations and architectures:** We evaluate the different masking configurations by training Zorro on AudioSet with a supervised loss and audio-visual contrastive loss. Specifically, we test the audio-visual trained models on a unimodal (Audio, Video) and multimodal setting. Our proposed configuration performs well across the board while providing additional unimodal outputs.

Architecture	Params	Fusion	Supervised (Audio+Video)			Self-Supervised (Audio+Video)		
			Video	Audio	Audio+Video	Video	Audio	Audio+Video
ViT	98M	Two Streams	23.1	40.1	42.2	18.9	32.3	34.8
ViT	98M	Input Level	9.1	31.6	42.2	Collapse	Collapse	Collapse
ViT	98M	Bottleneck [36]	9.7	32.6	42.5	Collapse	Collapse	Collapse
ViT	98M	Zorro	22.5	39.7	45.1	17.8	29.8	33.6
HiP	136M	Zorro	22.0	39.5	45.2	11.3	21.9	26.5
Swin	161M	Zorro	25.4	40.6	46.5	20.5	31.6	35.7

output suffer significantly from one missing modality. In contrast, both Zorro and using two separate modality streams achieve a high performance when only a single modality is provided. This is due to the fact that in those models, some capacity is allocated to each modality specifically and the model is able to produce unimodal outputs.

Unimodal pre-training for multi-modal fine-tuning:

Through the paper, we assumed availability of large multi-modal dataset for training. However, in some situations we only have available large amounts of unimodal samples (e.g. video or audio) and a small set of multi-modal data. To showcase the flexibility of our proposal, we run a single experiment where we train with two unimodal datasets and fine-tune on a smaller multi-modal dataset. We use only the audio signal from the AudioSet dataset and the videos from the Kinetics-400 dataset. When training, we mix batches with probability 0.5 per dataset, and do not compute the loss for the missing modalities. For evaluation, we fine-tune the resulting model on VGGSound and compare its result to the model trained from scratch. The fine-tuned model performs at 59.2 top-1 accuracy while the model trained from scratch performs at 54.4. This experiment shows the flexibility of the Zorro model to adapt to unimodal training while providing useful initialization for multi-modal fine-tuning.

4.5. Masking configurations

In this ablation, we study four different types of attention masking. First, we evaluate having data independent stream (*two streams*), where both models share weights but modalities are not connected. Secondly, we evaluate input level fusion, which consist of no masking in the model. This reduces the model to a vanilla ViT applied to the two concatenated modalities. Inspired by [36], we also evaluate *bottleneck masking* where the fusion tokens can attend to each modalities’ tokens but each modality can also attend to the fusion tokens. We want to make clear that although this approach uses the main proposal from MBT, it is not a reproduction of their work. This configuration forces each stream to mostly concentrate on one modality, but informa-

tion can flow across modalities through the fusion vectors. Finally, we compare all those masking strategies with our Zorro masking. For each masking configuration we train a model in a supervised manner (keeping the same number of outputs for fairness, except for the Two Streams which has two outputs). We also train the model in a self-supervised way, where the audio and the video outputs are used to compute the contrastive loss. To report performance, we train a linear classifier on top of the contrastive representations.

Table 3 reports the results. We extract two main conclusions. First, having modality independent streams is crucial for self-supervised training. Both the *input-level* and the *bottleneck* configurations immediately collapse as information can flow from one modality to the other. Performance for Zorro and *two streams* is very similar as Zorro when trained in a self-supervised manner reduces to the two stream architecture. Secondly, we find that having separate modality streams is useful also for supervised learning. Specially interesting is looking at the performances of *input-level*, *bottleneck* and Zorro, where Zorro performs better as the modality streams are more independently treated. We believe this is due to the ability of the model to keep modality-specific information through the network, which can be useful at later stages of processing. Finally, for self-supervised training of Zorro, we use equation 3, which trains also the fusion output. Although this produces a slight decrease on performance vs *two streams*, it’s beneficial for downstream tasks. Alternatively, when Zorro is trained using only audio and video outputs, it performs the same as *two streams* (35.0 vs 34.8) as the two models are equivalent.

5. Conclusion

In this paper, we introduced Zorro, a novel Transformer masking configuration which enables simultaneous unimodal and multimodal training and inference, as well as contrastive pre-training. Different from previous approaches to multi-modal perception, our proposed method is able to generate both unimodal and multimodal outputs. By splitting the information flow into unimodal and multimodal streams, we

are able to improve performance when the architecture is trained with a supervised loss and show the ability of the model to be self-supervised with a contrastive loss. We evaluate our model on multimodal tasks, showing great flexibility and state-of-the-art performance.

References

- [1] Sami Abu-El-Haija, Nisarg Kothari, Joonseok Lee, Paul Natsev, George Toderici, Balakrishnan Varadarajan, and Sudheendra Vijayanarasimhan. Youtube-8m: A large-scale video classification benchmark. *arXiv preprint arXiv:1609.08675*, 2016. [5](#), [7](#)
- [2] Hassan Akbari, Liangzhe Yuan, Rui Qian, Wei-Hong Chuang, Shih-Fu Chang, Yin Cui, and Boqing Gong. Vatt: Transformers for multimodal self-supervised learning from raw video, audio and text. *NeurIPS*, 2021. [2](#)
- [3] Jean-Baptiste Alayrac, Adrià Recasens, Rosalia Schneider, Relja Arandjelović, Jason Ramapuram, Jeffrey De Fauw, Lucas Smaira, Sander Dieleman, and Andrew Zisserman. Self-supervised multimodal versatile networks. In *NeurIPS*, 2020. [1](#), [2](#), [5](#)
- [4] Humam Alwassel, Dhruv Mahajan, Lorenzo Torresani, Bernard Ghanem, and Du Tran. Self-supervised learning by cross-modal audio-video clustering. In *NeurIPS*, 2020. [2](#)
- [5] Amir Amedi, Shir Hofstetter, Shachar Maidenbaum, and Benedetta Heimler. Task selectivity as a comprehensive principle for brain organization. *Trends in Cognitive Sciences*, 21(5):307–310, 2017. [1](#)
- [6] Relja Arandjelović and Andrew Zisserman. Look, listen and learn. In *ICCV*, 2017. [2](#)
- [7] Relja Arandjelović and Andrew Zisserman. Objects that sound. In *ECCV*, 2018. [2](#)
- [8] Anurag Arnab, Mostafa Dehghani, Georg Heigold, Chen Sun, Mario Lucic, and Cordelia Schmid. ViViT: A video vision Transformer. *CoRR*, abs/2103.15691, 2021. [2](#), [7](#)
- [9] Tadas Baltrušaitis, Chaitanya Ahuja, and Louis-Philippe Morency. Multimodal machine learning: A survey and taxonomy. *IEEE transactions on pattern analysis and machine intelligence*, 41(2):423–443, 2018. [2](#)
- [10] Daphne Bavelier and Helen J. Neville. Cross-modal plasticity: where and how? *Nature Reviews Neuroscience*, 3:443–452, 2002. [1](#)
- [11] Lukasz Bola, Maria Zimmermann, Piotr Mostowski, Katarzyna Jednoróg, Artur Marchewka, Paweł Rutkowski, and Marcin Szwed. Task-specific reorganization of the auditory cortex in deaf humans. *Proceedings of the National Academy of Sciences*, 114(4):E600–E609, 2017. [1](#)
- [12] Mathilde Caron, Hugo Touvron, Ishan Misra, Hervé Jégou, Julien Mairal, Piotr Bojanowski, and Armand Joulin. Emerging properties in self-supervised vision transformers. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 9650–9660, 2021. [6](#)
- [13] Joao Carreira, Skanda Koppula, Daniel Zoran, Adria Recasens, Catalin Ionescu, Olivier Henaff, Evan Shelhamer, Relja Arandjelovic, Matt Botvinick, Oriol Vinyals, et al. Hierarchical perceiver. *arXiv preprint arXiv:2202.10890*, 2022. [2](#), [4](#), [6](#), [12](#)
- [14] Joao Carreira and Andrew Zisserman. Quo vadis, action recognition? A new model and the Kinetics dataset. In *CVPR*, 2017. [2](#), [5](#), [7](#)
- [15] Honglie Chen, Weidi Xie, Andrea Vedaldi, and Andrew Zisserman. Vggsound: A large-scale audio-visual dataset. In *ICASSP 2020-2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 721–725. IEEE, 2020. [5](#)
- [16] Bowen Cheng, Ishan Misra, Alexander G Schwing, Alexander Kirillov, and Rohit Girdhar. Masked-attention mask transformer for universal image segmentation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 1290–1299, 2022. [2](#)
- [17] Dading Chong, Helin Wang, Peilin Zhou, and Qingcheng Zeng. Masked spectrogram prediction for self-supervised audio pre-training. *arXiv preprint arXiv:2204.12768*, 2022. [7](#)
- [18] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, et al. An image is worth 16x16 words: Transformers for image recognition at scale. *arXiv preprint arXiv:2010.11929*, 2020. [2](#), [3](#), [5](#)
- [19] Haytham M Fayed and Anurag Kumar. Large scale audiovisual learning of sounds with weakly labeled data. *arXiv preprint arXiv:2006.01595*, 2020. [2](#)
- [20] Christoph Feichtenhofer, Haoqi Fan, Jitendra Malik, and Kaiming He. Slowfast networks for video recognition. In *ICCV*, 2019. [7](#)
- [21] Christoph Feichtenhofer, Axel Pinz, and Andrew Zisserman. Convolutional two-stream network fusion for video action recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2016. [2](#)
- [22] Jort F Gemmeke, Daniel PW Ellis, Dylan Freedman, Aren Jansen, Wade Lawrence, R Channing Moore, Manoj Plakal, and Marvin Ritter. Audio set: An ontology and human-labeled dataset for audio events. In *2017 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 776–780. IEEE, 2017. [5](#)
- [23] Asif A. Ghazanfar and Charles E. Schroeder. Is neo-cortex essentially multisensory. *Trends in Cognitive Sciences*, 10(6):278–285, 2006. [1](#)

- [24] Yuan Gong, Yu-An Chung, and James Glass. Ast: Audio spectrogram transformer. *arXiv preprint arXiv:2104.01778*, 2021. [2](#), [7](#)
- [25] Kaiming He, Xinlei Chen, Saining Xie, Yanghao Li, Piotr Dollár, and Ross Girshick. Masked autoencoders are scalable vision learners. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 16000–16009, 2022. [6](#)
- [26] Andrew Jaegle, Sebastian Borgeaud, Jean-Baptiste Alayrac, Carl Doersch, Catalin Ionescu, David Ding, Skanda Koppula, Daniel Zoran, Andrew Brock, Evan Shelhamer, Olivier J. Hénaff, Matthew M. Botvinick, Andrew Zisserman, Oriol Vinyals, and João Carreira. Perceiver IO: A general architecture for structured inputs & outputs. *ICLR*, 2022. [3](#), [4](#), [12](#), [13](#), [14](#)
- [27] Andrew Jaegle, Felix Gimeno, Andrew Brock, Andrew Zisserman, Oriol Vinyals, and João Carreira. Perceiver: General perception with iterative attention. *ICML*, 2021. [1](#), [2](#), [4](#), [6](#), [12](#)
- [28] Will Kay, Joao Carreira, Karen Simonyan, Brian Zhang, Chloe Hillier, Sudheendra Vijayanarasimhan, Fabio Viola, Tim Green, Trevor Back, Paul Natsev, Mustafa Suleyman, and Andrew Zisserman. The kinetics human action video dataset. *arXiv preprint arXiv:1705.06950*, 2017. [5](#)
- [29] Evangelos Kazakos, Arsha Nagrani, Andrew Zisserman, and Dima Damen. Slow-fast auditory streams for audio recognition. In *ICASSP 2021-2021 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 855–859. IEEE, 2021. [7](#)
- [30] Bruno Korbar, Du Tran, and Lorenzo Torresani. Co-operative learning of audio and video models from self-supervised synchronization. In *NeurIPS*, 2018. [2](#)
- [31] Khaled Koutini, Jan Schlüter, Hamid Eghbal-zadeh, and Gerhard Widmer. Efficient training of audio transformers with patchout. *arXiv preprint arXiv:2110.05069*, 2021. [7](#)
- [32] Simon Lacey and K. Sathian. Visuo-haptic multisensory object recognition, categorization, and representation. *Frontiers in Psychology*, 5, 2014. [1](#)
- [33] Sangho Lee, Jiwan Chung, Youngjae Yu, Gunhee Kim, Thomas Breuel, Gal Chechik, and Yale Song. Acav100m: Automatic curation of large-scale datasets for audio-visual video representation learning. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 10274–10284, 2021. [5](#), [7](#)
- [34] Ze Liu, Han Hu, Yutong Lin, Zhuliang Yao, Zhenda Xie, Yixuan Wei, Jia Ning, Yue Cao, Zheng Zhang, Li Dong, et al. Swin transformer v2: Scaling up capacity and resolution. *arXiv preprint arXiv:2111.09883*, 2021. [2](#), [4](#), [6](#), [12](#)
- [35] Pedro Morgado, Nuno Vasconcelos, and Ishan Misra. Audio-visual instance discrimination with cross-modal agreement. *arXiv preprint arXiv:2004.12943*, 2020. [2](#)
- [36] Arsha Nagrani, Shan Yang, Anurag Arnab, Aren Jansen, Cordelia Schmid, and Chen Sun. Attention bottlenecks for multimodal fusion. *CoRR*, abs/2107.00135, 2021. [1](#), [2](#), [4](#), [5](#), [6](#), [7](#), [8](#), [12](#), [15](#)
- [37] Andrew Owens and Alexei A Efros. Audio-visual scene analysis with self-supervised multisensory features. In *ECCV*, 2018. [2](#)
- [38] Daniel S. Park, William Chan, Yu Zhang, Chung-Cheng Chiu, Barret Zoph, Ekin D. Cubuk, and Quoc V. Le. SpecAugment: A simple data augmentation method for automatic speech recognition. In *InterSpeech*, 2019. [6](#), [15](#)
- [39] Mandela Patrick, Yuki M. Asano, Ruth Fong, João F. Henriques, Geoffrey Zweig, and Andrea Vedaldi. Multi-modal self-supervision from generalized data transformations. *arXiv preprint arXiv:2003.04298*, 2020. [2](#)
- [40] Karol J Piczak. ESC: Dataset for environmental sound classification. In *ACM Multimedia*, 2015. [5](#)
- [41] Adrià Recasens, Pauline Luc, Jean-Baptiste Alayrac, Luyu Wang, Ross Hemsley, Florian Strub, Corentin Tallec, Mateusz Malinowski, Viorica Patraucean, Florent Alché, Michal Valko, Jean-Bastien Grill, Aäron van den Oord, and Andrew Zisserman. Broaden your views for self-supervised video learning, 2021. [2](#)
- [42] Tal Ridnik, Emanuel Ben-Baruch, Asaf Noy, and Lihi Zelnik-Manor. Imagenet-21k pretraining for the masses. *arXiv preprint arXiv:2104.10972*, 2021. [5](#), [7](#)
- [43] Arda Senocak, Tae-Hyun Oh, Junsik Kim, Ming-Hsuan Yang, and In So Kweon. Learning to localize sound source in visual scenes. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2018. [2](#)
- [44] Ladan Shams and Aaron R. Seitz. Benefits of multisensory learning. *Trends in Cognitive Sciences*, 12(11):411–417, 2008. [1](#)
- [45] Shinsuke Shimojo and Ladan Shams. Sensory modalities are not separate modalities: plasticity and interactions. *Current Opinion in Neurobiology*, 11(4):505–509, 2001. [1](#)
- [46] Nina Shvetsova, Brian Chen, Andrew Rouditchenko, Samuel Thomas, Brian Kingsbury, Rogerio Feris, David Harwath, James Glass, and Hilde Kuehne. Everything at once—multi-modal fusion transformer for video retrieval. *arXiv preprint arXiv:2112.04446*, 2021. [2](#)
- [47] Karen Simonyan and Andrew Zisserman. Two-stream convolutional networks for action recognition in videos. In *ICLR*, 2014. [2](#)
- [48] Alexandros Stergiou and Dima Damen. Play it back: Iterative attention for audio recognition. *arXiv preprint arXiv:2210.11328*, 2022. [6](#), [7](#)
- [49] Hugo Touvron, Matthieu Cord, Alaeldin El-Nouby, Jakob Verbeek, and Hervé Jégou. Three things ev-

- eryone should know about vision transformers. *arXiv preprint arXiv:2203.09795*, 2022. 12
- [50] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. In *NIPS*, 2017. 2, 3, 4
- [51] Sergey Verbitskiy, Vladimir Berikov, and Viacheslav Vyshegorodtsev. Eranns: Efficient residual audio neural networks for audio pattern recognition. *arXiv preprint arXiv:2106.01621*, 2021. 6, 7
- [52] Weiyao Wang, Du Tran, and Matt Feiszli. What makes training multi-modal classification networks hard? In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 12695–12705, 2020. 2
- [53] Fanyi Xiao, Yong Jae Lee, Kristen Grauman, Jitendra Malik, and Christoph Feichtenhofer. Audiovisual slowfast networks for video recognition, 2020. 2, 7
- [54] Shen Yan, Xuehan Xiong, Anurag Arnab, Zhichao Lu, Mi Zhang, Chen Sun, and Cordelia Schmid. Multi-view transformers for video recognition. *arXiv preprint arXiv:2201.04288*, 2022. 6
- [55] Bowen Zhang, Jiahui Yu, Christopher Fifty, Wei Han, Andrew M Dai, Ruoming Pang, and Fei Sha. Co-training transformer with videos and images improves action recognition. *arXiv preprint arXiv:2112.07175*, 2021. 6
- [56] Hongyi Zhang, Moustapha Cisse, Yann N Dauphin, and David Lopez-Paz. mixup: Beyond empirical risk minimization. *arXiv preprint arXiv:1710.09412*, 2017. 15

Appendix

In this appendix we expand the content of the paper in three different directions. First, we describe in detail the three Zorro architectures proposed in the paper and compare their speed. Second, we provide details about training, evaluation and input pre-processing. Third, we study the importance of the level at which the modalities are fused.

A. Architecture details

ViT: Zorro-ViT is based on the ViT-B/16 architecture, adapted for multi-modal processing (similar to [36]). For input video and audio frames, we use patch size 16×16 and hidden dimension of 768. The model has 12 self-attention layers, with intermediate dimension 3072 and 12 attention heads with $n = 6$ fusion tokens. For decoding cross-attention, we use a decoder with q_k dimension 1024. Figure 2 details the Zorro-ViT architecture. We use absolute learned position embedding.

Zorro-Swin: Swin [34] is a ViT-inspired transformer architecture which has shown improved efficiency and performance. The main innovation versus the original ViT architecture is to apply the self-attention operations on nearby tokens instead of all tokens in the input image. This reduces computational requirement while allowing the model to perform bottom-up inference. In order to build Zorro-Swin, we adapt the original 2D Swin [34] to deal with video data by adding a third dimension to the attention window (we use $3 \times 7 \times 7$) and the position encoding. Similarly to the 2D version, input tokens are only attended locally, and windows shift by half window size every two layers.

The Zorro-Swin architecture is depicted in Figure 3. As with Zorro-ViT, the model processes independently audio and video, while the fusion tokens cross-attend to the whole representation. The main difference is that to process video and audio, we use a Swin model. Specifically, to process the input audio (spectrograms) we use the original 2D Swin [34], while for video we use our 3D Swin architecture adaptation. For 2D Swin we use (4, 4) patches, while for 3D Swin we use (1, 4, 4). Furthermore, we use 6 fusion tokens (as in Zorro-ViT) and the input embedding dimension is 128. The number of layers per block is (2, 2, 6, 2) in both cases, number of heads are (4, 8, 16, 32), MLP dimensional ratio is 4 and we use stochastic layer drop with probability starting at 0 in the first layer and linearly increasing to 0.3 in the last layer. To process the fusion representation, we use the same self-attention as in Zorro-ViT with 16 heads and widening factor of 4. We use the relative position bias from [34]. Similarly to Zorro-ViT, no multimodal information flows into the unimodal streams.

Zorro-HiP: The hierarchical perceiver [13] extends the previously introduced Perceiver models [26, 27] models, by splitting the inputs into groups, and operating self-attention

only within those groups. Through the hierarchical architecture, those groups fuse together in order to aggregate information and globally reason about the input.

In our implementation of HiP, instead of using directly the pixels and audio signal as input, we create patches similarly to the ViT/Swin implementation. Specifically, inspired by [49], we produce the input patches by processing the input through a sequence of two (Convolution + LayerNorm + GeLU) operations and a final (Convolution + Layer Norm) at the end. The initial two convolutions project the input to 64 dimensions and the last one to 256 dimensions. The initial convolution has stride (2, 2, 2) when processing video and (1, 2, 2) when processing audio. The other convolutions have stride (1, 2, 2). The final downsample is (2, 8, 8) for video and (1, 8, 8) for audio. We add Fourier positional features and afterwards a learned embedding. Although the model does not need both positional features, we add the learned positional embedding to make the model as similar as possible to other Zorro variants.

In order to create Zorro-HiP, we use HiP building blocks for each modality and to process the multi-modal stream. Specifically, those blocks group the inputs into smaller sets, cross-attend using learned features and finally apply self-attention layers to the outputs of the cross attention operation (see [13] for more details). Figure 4 shows the Zorro-HiP architecture. Each modality is processed by a HiP model. Differently to the self-attention operation used Zorro-ViT and Zorro-Swin, in Zorro-HiP the multimodal tokens are processed by a HiP model. However, we skip the initial block of HiP for the fusion latents as these operations would only be applied to learned embeddings. After the three blocks are processed, we concatenate all of them together in order to create the input for the next level of the fusion stream, which splits the input in groups and cross attend to them all together. The three HiP models are sharing the same weights and architecture. We start with 6 embedding vectors for the fusion stream, drop layers with probability 0.1. The number of self-attend per blocks are (2, 2, 2, 12, 2), the number of groups per block is (32, 4, 1, 1, 1), the number of latent vectors per group in each block is (128, 256, 256, 256, 256), the number of channels per block is (256, 256, 512, 512, 1024), the number of heads per block is (8, 8, 16, 16, 32). Finally, we using widening factor of 4 in self-attention.

Training speed: For most of the experiments we use 128 TPU-v3 cores. Here, we compare the speed of the three presented Zorro architectures. Under similar conditions (8 frames at 3.12 FPS) and using batch size 512, Zorro-ViT (98M parameters) has a speed of 3.2 steps per second, Zorro-Swin (161M parameters) has a speed of 4.2 steps per second and Zorro-HiP (136M parameters) has a speed of 5.4 steps per second, where each step is a forward and backward pass. For Zorro-ViT, when we use separate weight per each stream (267M parameters), the speed drops to 2.8sps. Zorro-HiP

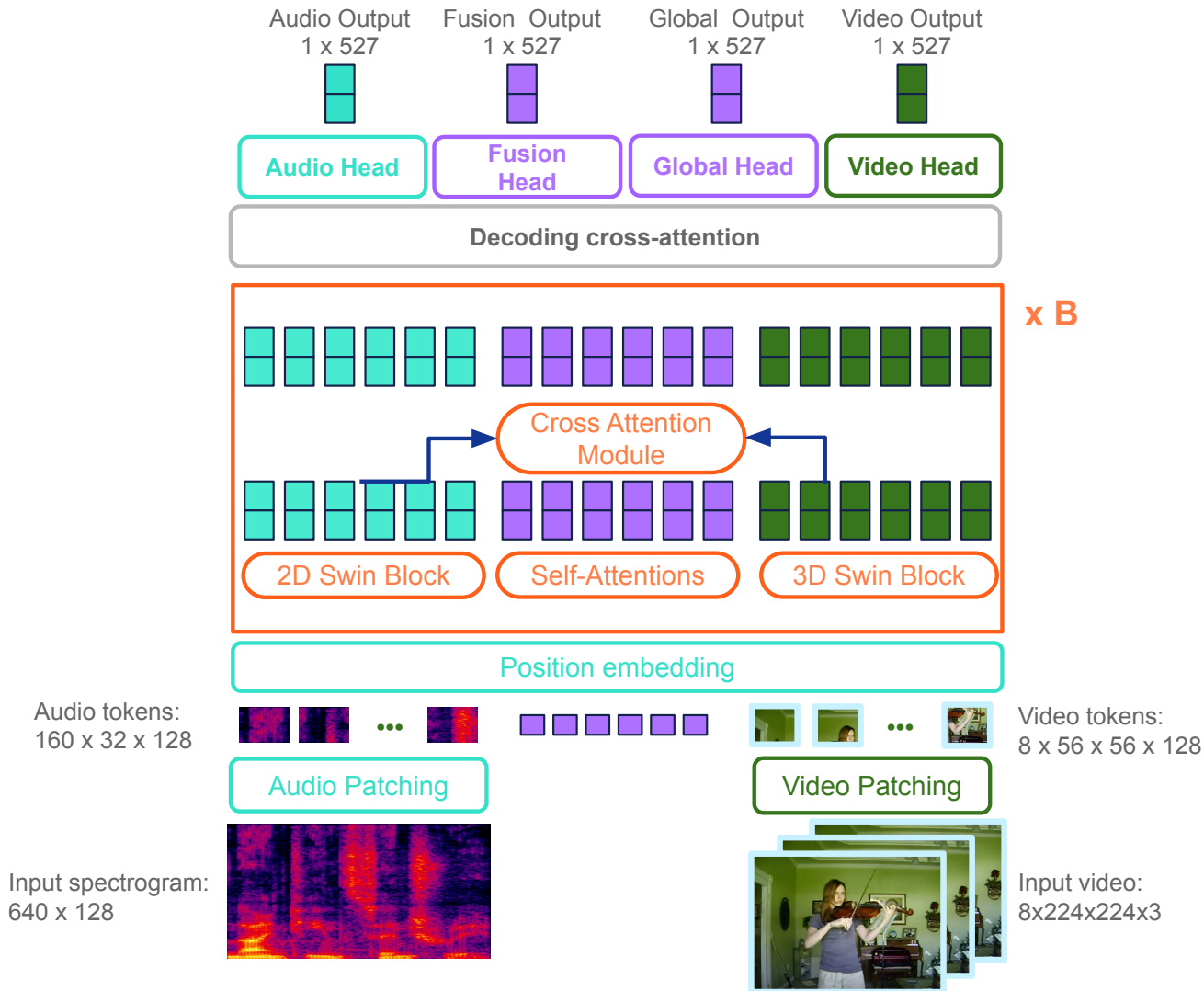


Figure 3. **Zorro-Swin**: The input to our model are video frames and audio spectrograms. Each of those inputs is patched using a 2D convolution and projection to input dimension $D = 128$. Next, the audio tokens are processed by a 2D Swin, while the video tokens are processed by a 3D Swin. The fusion tokens are processed by standard self-attention layers. At the end of each of the B blocks, a cross-attention operation is applied to produce the next fusion tokens. Specifically, our architecture blocks the information to flow towards the unimodal hidden representation, while still allowing the fusion representation to access all modalities. By doing this, we ensure that the video and audio representations have gated access to (i.e. depend on) only the video and audio inputs respectively. To produce the outputs, following Perceiver IO [26], we learn a set of queries that cross-attend to the unimodal and multimodal representation. We also use masking at the decoding stage to make sure we can produce unimodal outputs as well as multimodal outputs. By doing this, we can train Zorro-Swin using a self-supervised loss which requires unimodal representations.

and Zorro-Swin are clearly faster models than ViT. However, as reported in Table 2 in the paper, when fine-tuning, Zorro-ViT still performs the best of the three architectures.

B. Implementation and training details

Audio and video pre-processing

The inputs to our model are video and audio. The audio and video data are synced and cover the same time span.

Video augmentation: The input videos consist of 8 frames of size 224×224 . When training on AudioSet from scratch

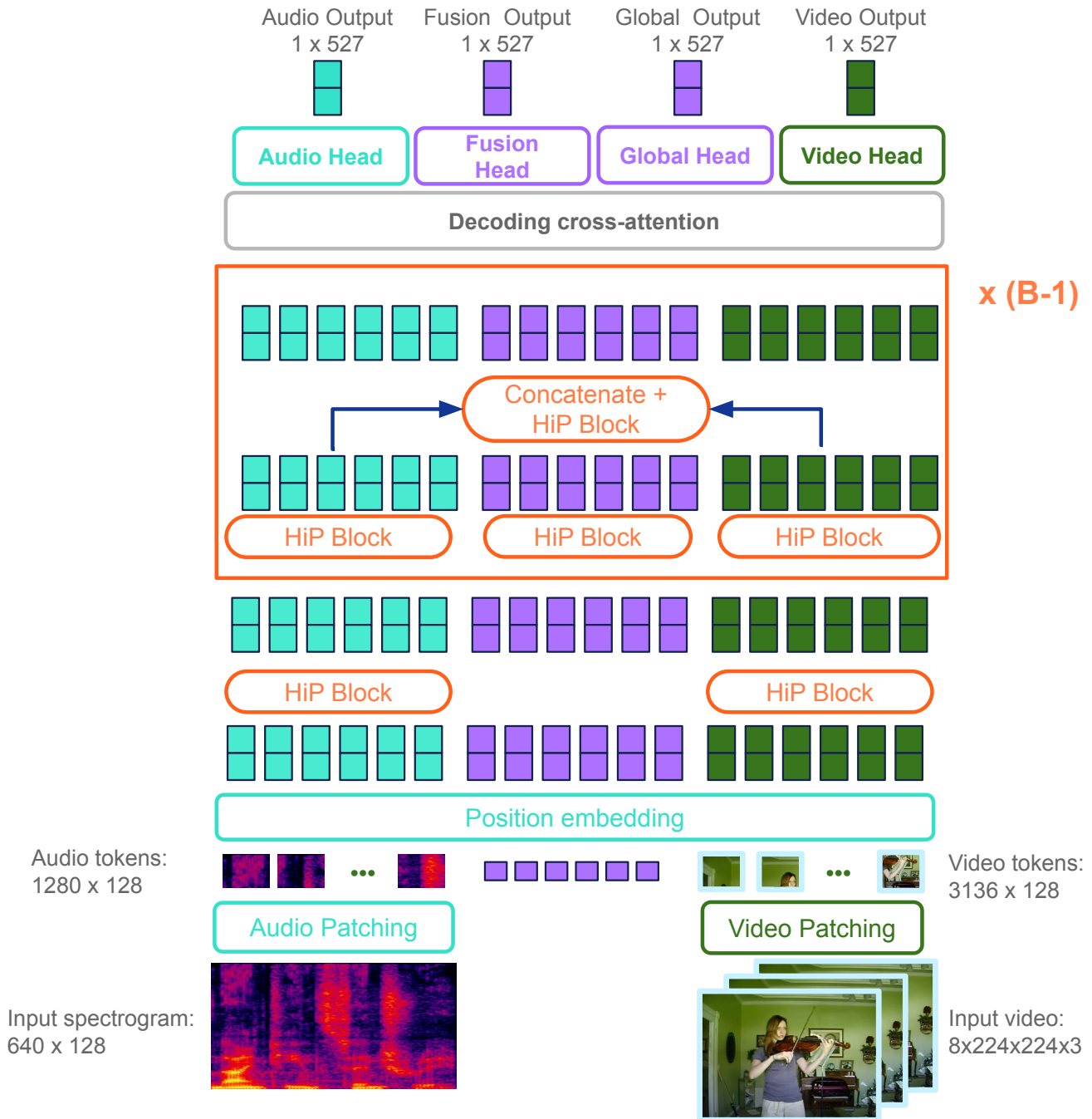


Figure 4. **Zorro-HiP**: The input to our model are video frames and audio spectrograms. Each of those inputs is patched using a sequence of 3D-convolutions and projection to input dimension $D = 256$. Next, each modality and the fusion tokens are processed through independent HiP blocks (which share weights). The architecture blocks the information from flowing towards the unimodal hidden representation, while still allowing the fusion representation to access all modalities. By doing this, we ensure that the video and audio representations have gated access to (i.e. depend on) only the video and audio inputs respectively. To produce the outputs, following Perceiver IO [26], we learn a set of queries that cross-attend to the unimodal and multi-modal representation. We use masking at the decoding stage to make sure we can produce unimodal outputs as well as multimodal outputs. By doing this, we can train Zorro using a self-supervised loss which requires unimodal representations.

or fine-tuning in Kinetics, we sample videos at 3.12FPS. When training in YouTube-8M, ACAV-100M or fine-tuning in VGGSound we sample at 1.25FPS. Finally, when fine-tuning on AudioSet-500k or training Zorro-HiP in ACAV-100M, we sample at 1FPS. During training, we use random cropping as well as color augmentation in the visual input. For random cropping, we sample a random bounding box which covers at least 30% of the frame and the whole frame at maximum, with a random aspect ratio in the range (0.9, 1.33). We apply color augmentation with probability 0.8, color randomisation on saturation and contrast (0.6, 1.4), brightness (max_delta=32/255) and hue with (max_delta=0.2).

Audio augmentation: The audio input consist of 2.56s for training from scratch on Audioset and fine-tuning in Kinetics, and 6.4s when pre-training on YouTube-8M and ACAV-100M or finetuning in ESC-50 and VGGSound and 8s when fine-tuning in AudioSet or training Zorro-HiP in ACAV-100M. Audio is sampled at 48kHz. We use Log-Mel spectrograms as inputs to our model using 128 bins with Hanning windows of length 1200 samples and stride 480. To augment the audio, we use SpecAugment [38] and frequency jittering where we shift the frequency by an integer sampled on the range (-10, 10). When fine-tuning ESC-50, we use the same input length as in pre-training (6.4s). As the input length is larger than ESC-50 samples, we loop over the samples.

Evaluation details: For evaluation, we use equally spaced 8 frame clips with the same stride as during training. Furthermore, we take 3 crops for each of those clips and average the resulting predictions for multi-crop evaluation. When evaluating ESC-50, we feed a single clip as it covers the whole length of the signal. We report performance in the test set for AudioSet and VGGSound and validation set for Kinetics-400. For ESC-50, we average the performance of the 5 splits.

Optimisation details: Details about optimizers and hyperparameters for models used in the paper can be found in Table 5. For AudioSet from scratch we select the highest learning rate (using 0.0003 as maximum) that does not lead to collapse. When fine-tuning, we choose the learning rate by evaluating without augmentation. In ESC-50, we use the first split to select the learning rate. We train all our models with batch size 512 and we use learning rate scaling with factor $\frac{batch\ size}{256}$. We train all models for 50 epochs except for the ACAV-100M models which we train for 10 epochs. For the ablations *Bottleneck* and *Vanilla*, we train for 25 epochs to prevent overfitting. For fine-tuning in AudioSet-500k, we use label smoothing of 0.15 and modality mixup [56]. Different from [36], we need to provide supervision not only for the multimodal output but also for the unimodal output. This means the mixup procedure can be performed in many

Table 4. **Modality fusion position.** We report the performance of our models trained on audio-visual data when the multi-modal fusion is done at different layers.

Fusion level	Audio	Video	Audio+Video
0	37.6	20.8	44.2
3	38.0	21.4	44.6
6	37.5	20.3	44.2
9	37.6	20.9	44.4

different ways. We find sampling a single mixup value from a $\beta(0.3, 0.3)$ the best configuration to apply mixup.

Contrastive learning: In order to pre-train Zorro using the audio-visual contrastive loss, we define two projectors for the audio and video outputs of the model, with different weights. When training the model with the complete contrastive loss involving the fusion weights, we create a third projector for the fusion output. Those projectors consist of an MLP using hidden dimensionality of 512. We use temperature $\tau = 0.08$ for the training loss.

C. Fusion position

In this section, we extend the study presented in the main paper to study the effect of introducing the fusion tokens starting at a certain level of the network, inspired by [36]. For these experiments, we trained our models on AudioSet-2M for 25 epochs and report performance using the standard evaluation protocol as described in Section B.

In Table 4 we report the performance of our model when introducing the fusion stream at a different layer of the network. In previous layers, the fusion tokens are not used. Interestingly, the position of where to introduce the fusion layer does not seem critical to the overall performance of the model. We attribute this to the fact that Zorro keeps its unimodal streams untouched, preventing the full representation from overfitting to the most informative modality for a given task. In order to align with standard architectures, we choose to use our fusion layers from the beginning of the model.

Table 5. **Hyperparameters.** For reproducibility, in this table we report the hyperparameters used for each model in the paper.

Model	Sup.	Dataset	Scratch/Fine-tuning	Optimizer	Learning Rate	Weight Decay
Zorro-ViT	Supervised	AS	Scratch	Adam	0.0003	10^{-6}
Two Streams	Supervised	AS	Scratch	Adam	0.0001	10^{-6}
Vanilla Fusion	Supervised	AS	Scratch	Adam	0.0001	10^{-6}
Bottleneck Fusion	Supervised	AS	Scratch	Adam	0.0001	10^{-6}
Zorro-ViT	Contrastive	AS	Scratch	Adam	0.00005	10^{-6}
Two Streams	Contrastive	AS	Scratch	Adam	0.00005	10^{-6}
Zorro-Swin	Supervised	AS	Scratch	Adam	0.0001	10^{-6}
Zorro-HiP	Supervised	AS	Scratch	Adam	0.0001	10^{-6}
Zorro-Swin	Contrastive	AS	Scratch	Adam	0.0001	10^{-6}
Zorro-HiP	Contrastive	AS	Scratch	Adam	0.00001	10^{-6}
Zorro-ViT	Supervised	YT8M	Scratch	Adam	0.00008	10^{-6}
Zorro-ViT	Contrastive	ACAV	Scratch	Adam	0.00005	10^{-6}
Zorro-Swin	Contrastive	ACAV	Scratch	Adam	0.00005	10^{-6}
Zorro-HiP	Contrastive	ACAV	Scratch	Adam	0.0001	10^{-6}
Zorro-ViT	Supervised	AS-500k	FT (ACAV-100M)	SGD	0.08	10^{-6}
Zorro-ViT	Supervised	AS-500k	FT (YT-8M)	SGD	0.06	0
Zorro-ViT	Supervised	AS-500k	FT (IN-21k)	SGD	0.1	0
Zorro-Swin	Supervised	AS-500k	FT (ACAV-100M)	SGD	0.05	0
Zorro-HiP	Supervised	AS-500k	FT (ACAV-100M)	Adam	0.0001	0
Zorro-ViT	Supervised	VGGSound	FT (ACAV-100M)	SGD	0.01	0
Zorro-ViT	Supervised	VGGSound	FT (YT-8M)	SGD	0.01	0
Zorro-ViT	Supervised	VGGSound	FT (IN-21k)	SGD	0.08	0
Zorro-Swin	Supervised	VGGSound	FT (ACAV-100M)	SGD	0.05	0
Zorro-HiP	Supervised	VGGSound	FT (ACAV-100M)	Adam	0.0001	0
Zorro-ViT	Supervised	K400 (V+A)	FT (ACAV-100M)	SGD	0.05	0
Zorro-ViT	Supervised	K400 (V+A)	FT (YT-8M)	SGD	0.07	0
Zorro-ViT	Supervised	K400 (V+A)	FT (IN-21k)	SGD	0.08	0
Zorro-Swin	Supervised	K400 (V+A)	FT (ACAV-100M)	SGD	0.05	0
Zorro-HiP	Supervised	K400 (V+A)	FT (ACAV-100M)	Adam	0.0001	0
Zorro-ViT	Supervised	K400 (V)	FT (ACAV-100M)	SGD	0.08	0
Zorro-ViT	Supervised	K400 (V)	FT (YT-8M)	SGD	0.08	0
Zorro-ViT	Supervised	K400 (V)	FT (IN-21k)	SGD	0.08	0
Zorro-Swin	Supervised	K400 (V)	FT (ACAV-100M)	SGD	0.05	0
Zorro-HiP	Supervised	K400 (V)	FT (ACAV-100M)	Adam	0.0001	0
Zorro-ViT	Supervised	ESC-50	FT (ACAV-100M)	Adam	0.0009	0.001
Zorro-ViT	Supervised	ESC-50	FT (YT-8M)	Adam	0.0009	0.001
Zorro-ViT	Supervised	ESC-50	FT (IN-21k)	Adam	0.0009	0.001
Zorro-Swin	Supervised	ESC-50	FT (ACAV-100M)	Adam	0.0007	0
Zorro-HiP	Supervised	ESC-50	FT (ACAV-100M)	Adam	0.0003	0