NAS just once: Neural Architecture Search for joint Image-Video Recognition

Sofia Casarin[1], Sergio Escalera[2,3], Oswald Lanz[1]
[1]Free University of Bozen-Bolzano, Bolzano, Italy
[2]Computer Vision Center, Barcelona, Spain
[3]Universitat de Barcelona, Barcelona, Spain
scasarin@unibz.it, sergio@maia.ub.es, lanz@inf.unibz.it

Abstract

Neural Architecture Search (NAS) for Video Understanding has slowly advanced compared to the Image-domain counterpart. Current approaches often focus on 3D networks, search for untied spatial and temporal components, or for pseudo-3D operators. As NAS methods for image-related tasks are often unsuitable for videos due to the lack of benchmarks like NASBench-101, many video-NAS methods use naïve search procedures and fail to leverage advancements in search mechanisms developed for NAS for image tasks. In this work, we propose the first approach to bridge the gap between NAS for Videos and IMages (VIM-NAS), proposing a unique solution to find high-performing and efficient neural networks across ImageNet, Kinetics-400, Kinetics-600, and Something-SomethingV2 datasets. We optimize the 2D space and 3D space-time tubes to tokenize images and videos, along with the architecture of a unique supernet Vision transformer, via a differentiable weight-entanglement mechanism. Leveraging a multi-dataset training strategy, VIM-NAS achieves 84.4% Top-1 accuracy on ImageNet, 90.7% on Kinetics-400, improves state-of-the-art on Kinetics-600 by 0.4%, and improves previous NAS-SOTA by 13.4% on Something-SomethingV2 reducing the accuracy gap with hand-designed neural networks in Video Action Recognition.

1. Introduction

Neural Architecture Search (NAS) has achieved state-of-the-art (SOTA) accuracy - efficiency trade-off for CNNs [3, 7, 55]. During years many efforts were made to make NAS efficient, as it is naturally defined as a computational intensive bi-level optimization problem [17, 30, 61], reducing in many cases the runtime of the NAS pipeline to less than 24 GPU hours. Given these achievements, recent directions focus on a further step: improving the generalization abilities of NAS to avoid re-running the same procedure for every new dataset [23, 58, 68]. While great progress has been made in NAS for Image Understanding, the research has advanced slowly in NAS for Video Understanding [20, 60]. As designing DNNs for videos requires high domain expertise, advancing video NAS would facilitate video model design, enhance model compression for memory-intensive video data, and improve performance in this increasingly popular field. Moreover, video NAS could address open questions *e.g.* the impact of resizing images to match ImageNet object sizes, the performance differences between 2D and related 3D backbones, and the importance of temporal *vs* appearance features. As in the literature many of the proposed video models are built on top of image ones, related NAS approaches have similarly addressed the problem, proposing NAS methods where i) 2D is replaced with a 3D convolution [20, 72], ii) the spatial and the temporal components are searched untied [36, 40, 60] generally with a naïve DARTS [30] approach – proved not to perform well when directly applied to search temporal components [60], iii) the search space is composed of pseudo-3D operators [35]. Moreover, among the existing approaches tailoring NAS for video action recognition, only two works [20, 60], at the time of writing, address datasets where the temporal component is highly relevant to recognize the action, as in Something-SomethingV2 dataset [16]. After examining how NAS for video action recognition has been addressed and considering the significant effort in the NAS community to develop generalizable approaches for images, a natural question arises: *Why should we re-run the procedure when going from Image to Video Understanding? Can we have a unique NAS framework solving both image classification and video action recognition?*

The problem is complex, as many NAS methods designed for the image domain are either not straightforward for videos, *e.g.* differentiable methods [6, 30], or cannot be used, *e.g.* fast predictor-based algorithms [61]. Indeed, the former leads to not-comparable results to hand-designed neural networks, as the pipeline to pre-train the one-shot supernetwork on ImageNet is not clear: should we perform half-search on ImageNet and half on the video dataset? Should we perform the search on the video dataset and pre-train only the found architecture? The latter cannot

Table 1. Summary of our approach properties compared to leading methods in NAS for Video Action Recognition, Image Classification, and Vision Transformers.

	EvaNET [36]	AutoX3D [20]	NAS-TC [40]	AutoFormer [62]	MDL-NAS [58]	**Ours**
Image	✗	✗	✗	✓	✓	✓
Video	✓	✓	✓	✗	✗	✓
Motion Relevant	✗	✓	✗	✗	✗	✓
Scene Relevant	✓	✓	✓	✗	✗	✓
Complexity (N = # datasets)	$O(N)$	$O(N)$	$O(N)$	$O(N)$	$O(1)$	$O(1)$
Differentiable	✗	✗	✓	✗	✗	✓

be used as it would require benchmarks like NAS-Bench-101/201 [66], [10], built in many years, that provide labels for predictors. To tackle the problem, we develop a NAS framework optimizing a Vision Transformer (ViT) [11] architecture and the input tokenization (through 2D space and 3D space-time tubes) to properly handle both images and videos (Fig. 1). Transformers, being extremely data-hungry, have already proved to benefit from the joint-training of images and videos [27, 38, 71]. Furthermore, despite their significant achievements, the architectural design of these models remains underexplored. Current ViTs partition an image into a series of tokens which are then processed by stacked transformer blocks, as in NLP tasks. However, this straightforward approach does not necessarily guarantee optimal performance for vision tasks [45]. For example, the internal structure of the blocks should be better investigated, including aspects like patch/tube size of the input, the number of heads in multi-head self-attention (MHSA), output dimensions of parametric layers, operation types, and the overall depth of the model.

In this work, we propose a new architecture search algorithm, named VIM-NAS, that optimizes in one NAS-run pure vision transformer models for two input modalities: images and videos. Without the need to run the NAS search for every new dataset/task, our approach is the first, to the best of our knowledge, to propose a unified framework solving image and video tasks (refer to Tab. 1 for a summary of the properties of VIM-NAS compared to previous works). As ViT with tube sampling can efficiently process both modalities, our unified framework benefits the NAS domain by reducing search time, enhancing task-specific performance (as demonstrated empirically), and provides a basis in understanding the natural link between image and video models. Our approach primarly tackles two issues in the optimal ViT search. 1) How to properly tokenize different input modalities –ImageNet vs Kinetics–, and input with different inductive biases –scene relevant Kinetics vs motion relevant Something–. 2) How to properly train the supernetwork and the subnetworks on multiple datasets. We build a large search space (summarized in Tab. 2) that includes the input tokenization, the embedding dimension, the number of heads, the query/key/value dimension, the MLP ratio, and the network depth. We deploy a supernet training strategy using a differentiable weight entanglement mechanism under no resource-constraints, and an evolutionary algorithm under resource-constraints. The training is regularized by an additional loss term that measures the distribution gap between one-shot and stand-alone models. The tokenization of the input is optimized by finding 2D and 3D tube shapes. Our trained supernet produces without extra fine-tuning or retraining from scratch extremely high-quality networks, achieving 84.4 % Top-1 accuracy on Imagenet, 90.7 % on Kinetics-400, 92.2 % on Kinetics-600, and 76.9 % on Something-Something-v2, with much fewer parameters compared to hand-designed models. In summary, we make three major contributions in this paper:

1. To the best of our knowledge, this work is the first effort to design a NAS algorithm to solve both Image Classification and Video Action Recognition.
2. We propose a simple yet effective framework for properly handling co-training of the supernetwork on multiple datasets to achieve high performance on all datasets and deploy a differentiable weight entanglement mechanism that ensures efficient well-trained subnetworks. Different from previous approaches, our setup is comparable to hand-designed networks as we can exploit pre-training.
3. We optimize tube shapes for images and videos, showing how processing the input impacts network performance. We obtain the *first comparable* performance to hand-designed state-of-the art results through NAS on the challenging Something-Something v2 dataset, Kinetics-400, and Kinetics-600.

2. Related Works

We briefly review the state-of-the-art in video action recognition (Sec. 2.1), previous NAS methods for video action recognition (Sec. 2.2), and describe works strictly related to ours focusing on NAS for ViTs (Sec. 2.3).

2.1. Video Action Recognition

In the past few years, 3D-based CNNs [4, 13, 39, 41, 52, 63] have replaced 2D CNNs [21, 28, 43] in the field of video action recognition. Two streams of image architectures have

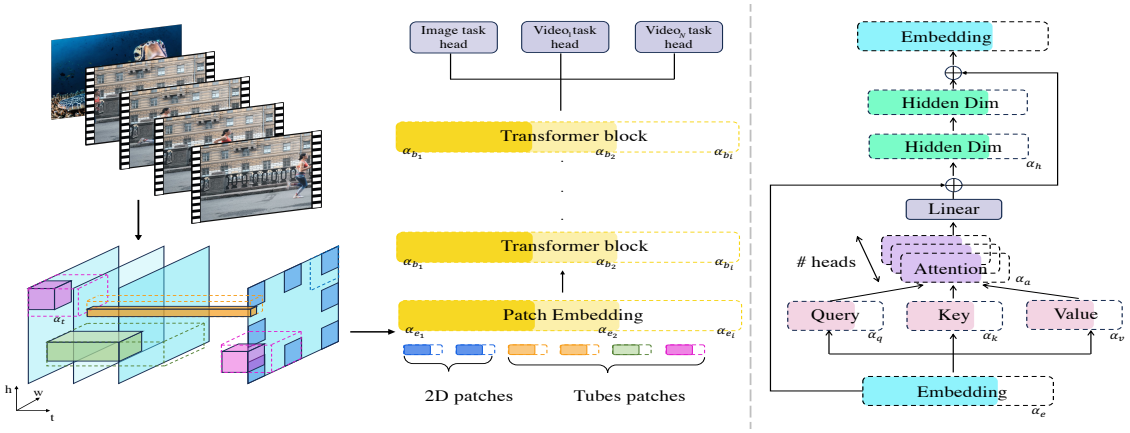

Figure 1. **Left**: The overall components of the VIM-NAS supernet, which processes images and videos through dynamic 2D and 3D sparse overlapping tubes. The search space includes different tube shapes (highlighted by different colors), dynamic transformer blocks in each layer and depths that are trained through weight entanglement, as indicated by the different shades. Solid lines refer to chosen components while dashed lines to optimized ones through the α parameters of the differentiable formulation. **Right**: The detailed ViT block where the embedding dimension, number of heads, MLP ratio, Q-K-V dim are optimized. Please refer to Sec. 3.2 for additional information.

been used by Two-Stream Network [59] to handle single-frame and multi-frame optical flow inputs. I3D [4] has expanded the Inception architecture to more costly 3D convolutions, while R(2+1)D [53] has added 1D temporal convolution to the 2D ResNet model to capture temporal feature. In order to simulate long-term temporal relations, LSTMs are applied to image features that are extracted from video frames [9, 32]. Computationally cheaper alternatives were proposed through efficient blocks like Temporal Shift Module (TSM) [28] and Gate Shift Module (GSM) [47] to model the temporal dynamics. Since the advent of transformer models and self attention [54], ViTs have also been deployed for videos, requiring additional components, such as sparse sampling that treats videos as tubes [38, 57], or space-time factorized attention [1, 2, 64].

2.2. NAS for Video Action Recognition

NAS has achieved highly competitive performance in image classification tasks [29, 73]. However, there are not many methods to combine video action recognition with NAS. The main reason is that NAS has high requirements for computation resources, and lightweight approaches as predictor-based ones [61] would require a NAS benchmark to be trained on. EvaNet [36], the first proposed approach in the field, AssembleNet [41], and Tiny Video Networks [37] all exploit evolutionary algorithms as a search strategy, targeting video datasets where the temporal component is not relevant to recognize the action. AutoX3D [20] and PV-NAS [60] on the other hand, address the Something-SomethingV2 task, but while the former searches computational intensive 3D architectures, the latter uses a naïve DARTS approach to look for spatial and temporal cells, showing eventually how naïve DARTS cannot be used directly to look for video networks. NAS-TC [40] does not target Something-SomethingV2, and separates the search phase by looking independently for the spatial and the temporal cells, using DARTS. In [35] the authors propose a DARTS approach to look for pseudo-3D operators, validating the method on UCF-101 dataset, and finally in [72] 3D Resnet-like architectures are searched through a differentiable approach. Differently from previous approaches, we address the problem to propose a unified framework, able to solve both image classification and video action recognition, including in our evaluation Something-SomethingV2.

2.3. NAS for ViT

Our work is closely related to Autoformer [62], which introduced weight entanglement to restrict memory consumption on one-shot NAS, to GraViT-E [48], which proposes the differentiable version, and to ViTAS [45], which introduces a cyclic weight sharing mechanism. All methods address the problem of avoiding gradient conflicts among subnetworks, typically experienced in one-shot approaches. While Autoformer can be seen as the baseline, GraViT-E re-frames the problem in a continuous search space, and ViTAS discards the Autoformer ordinal weight sharing in favor of a new weight sampling strategy to allow subnetworks to be more fairly trained. Our approach extends upon these, as we deploy a one-shot differentiable approach with a weight entanglement mechanism. However, we integrate a distribution consistent constraint to address the inconsistency between the weights that are trained with a stand-alone model and the ones that are inherited [33], as already proven in [45] that the ordinal mechanism adopted by [62] and [48] introduces imbalance among channels during training, inducing sub-optimal architectures. Moreover, differently from previous methods, we extended this not only to the ViT architecture but also to the 2D and 3D convolutions that define the tube shapes, providing an optimized input tokenization crucial for properly handling different input modalities with different inductive biases.

3. Method

In this section, we introduce the main components of our method. Sec. 3.1 defines the one-shot NAS problem and gives details on the weight-entanglement mechanism in a differentiable formulation. In Sec. 3.2 we define our NAS pipeline, consisting of the search-space definition, the joint-training strategy, and the evolutionary algorithm we deployed to meet different resource constraints.

3.1. Differentiable One-Shot NAS with Weight Entanglement

One-shot NAS Problem Definition Most existing NAS methods are formulated as a constrained optimization problem, where the optimal architecture α^* is searched from a search-space summarized, in one-shot approaches, in the supernetwork $\mathcal{A}$. Usually, to avoid exhausting path training from scratch, one-shot approaches leverage a weight-sharing strategy. Each operator in the search space has an associated weight matrix $\boldsymbol{w}$, which is shared across the network, *i.e*. all architectures that include a specific operator utilize the same shared weight matrix. A subnetwork α inherits its weights from the corresponding supernetwork weight matrix $\boldsymbol{w}_{\mathcal{A}}$. Given an operation with a set $\mathcal{C}$ of candidate dimensions, where $c \in \mathcal{C}$ denotes the dimensions within α, the optimization function is:

$$\begin{aligned} \alpha^* &= \underset{\alpha \in \mathcal{A}}{\operatorname{argmin}} \mathcal{L}_{val}(\alpha, \boldsymbol{w}^*_{\alpha}) \quad \text{s.t.} \\ \boldsymbol{w}^*_{\mathcal{A}} &= \underset{\boldsymbol{w}_{\alpha}}{\operatorname{argmin}} \mathcal{L}_{train}(\mathcal{A}, \boldsymbol{w}_{\mathcal{A}}) \quad \text{with} \quad g(\alpha) \leq f, \end{aligned} \tag{1}$$

where the superformer $\mathcal{A}$ depends on the maximum dimensions $\mathcal{C}$, and $\alpha^* = \mathcal{A}(c^*)$, $\alpha = \mathcal{A}(c)$ are obtained by sampling with the specified dimension from $\mathcal{A}$. Eq. (1) aims at minimizing the validation loss of the best network α^*, by minimizing the training loss of $\mathcal{A}$ under a resource budget f computed from the function $g(\cdot)$. As this formulation requires running the procedure F times for F different resource budgets, we also implemented an evolutionary search after training the supernetwork with our differentiable formulation, initially without constraints. This was necessary for preliminary experiments involving multiple resource constraints, as the evolutionary search takes approximately 5 minutes per constraint. We provide a brief overview of the evolutionary search in Sec. 3.2, with detailed algorithmic information available in the Supplementary Material. Afterwards, we selected a specific constraint and executed the complete procedure according to Eq. (1).

Weight Entanglement Mechanism Weight-sharing, commonly adopted in one-shot NAS approaches, comes with a set of advantages and disadvantages. On one hand, it allows a large number of models to be trained and searched at the expense of training a single supernet. On the other hand, the memory requirements for training the supernet, increases linearly with the number of operator choices per edge, making it infeasible to train very large models. Therefore, we deploy *weight-entanglement*, which goes one step further. While *weight-sharing* shares the operator parameters between different architectures, weight-entanglement shares the parameters between different operators on a given edge of the supernet. For example, the parameters of a smaller convolutional kernel are sampled from the largest convolutional kernel on the same edge. Specifically, given a subnetwork $\alpha \in \mathcal{A}$ with l layers, its structure and weights are represented as: $\alpha = (\alpha^{(1)}, \ldots, \alpha^{(i)}, \ldots, \alpha^{(l)})$, and $w = (w^{(1)}, \ldots, w^{(i)}, \ldots, w^{(l)})$, where $\alpha^{(i)}$ represents the sample block in the i-th layer and $w^{(i)}$ the corresponding weights. Each layer is characterized by multiple choices of blocks, and during the search-phase $\alpha^{(i)}$ and $w^{(i)}$ are selected from a set of n possible blocks that belong to the search space. Thus:$\alpha^{(i)} \in \{b_1^{(i)}, \ldots, b_j^{(i)}, \ldots, b_n^{(i)}\}$, $w^{(i)} \in \{w_1^{(i)}, \ldots, w_j^{(i)}, \ldots, w_n^{(i)}\}$, where $b_j^{(i)}$ is a candidate block in the search space and $w_j^{(i)}$ are its weights. For any two blocks $b_j^{(i)}$ and $b_k^{(i)}$ the weight-entanglement strategy makes the weight updates entangled with each other by imposing: $w_j^{(i)} \subseteq w_k^{(i)}$ or $w_k^{(i)} \subseteq w_j^{(i)}$.

Differentiable Formulation Our approach exploits the differentiable formulation proposed in [48] to combine weight-entanglement with gradient-based NAS. Similar to weight-entanglement, the operators on any given edge reuse the parameters of the largest operator on the edge, and each operator learns the architectural parameters, similar to gradient-based one-shot NAS. Differently from Autoformer [62], the optimal architecture is found by directly discretizing the supernet using the learned architectural parameters (α in Fig. 1). At the end of the search phase, we *do not* choose the optimal operations according to the largest parameters, as done in [30]. DARTS, as demonstrated in [56], is based on the incorrect assumption that the parameter which expresses the strength of an operation correlates with the discretization accuracy at convergence. We instead utilize a perturbation-based methodology, wherein the importance of each operator is assessed based on its impact on the performance of the neural network. To achieve this, we systematically mask each operator on a specific edge while preserving the others, and re-evaluate the superformer. The operator that leads to the most substantial drop in network validation accuracy is recognized as the essential operation associated with that edge.

3.2. Search Pipeline

Search Space Definition. To explore the optimal ViT architecture, our transformer space incorporates the *tube-shapes, embedding and Q-K-V dimensions, number of heads, MLP ratio, and network depth*, as detailed in Fig. 1 and in Tab. 2. Our search-space includes the tubes-shapes

as, similarly to [38], given a video $V \in \mathcal{R}^{T\times H\times W\times C}$, and an image $I \in \mathcal{R}^{H\times W\times C}$, images are processed by 2D kernels, and videos are processed by sparse overlapping tubes defined by 2D and 3D kernels. These tubes may have different shapes, *e.g.* $32 \times 4 \times 4$ to capture long temporal information at low spatial resolution, and may have different strides. We include searched offsets to pick the start location of a tube and ensure the starting location is not always (0,0,0). The computational cost is therefore reduced, as the sparse spatial and temporal sampling limits the number of video-specific tokens to a maximum determined by our search-space definition. Similar to [38], we employed a fixed sine/cosine embedding. We include the stride, kernel shape, and offset of each tube to guarantee that its positional embedding has the global spatio-temporal location. Given $\tau = 10000$, similarly to [11], for each dimension $i = (t, x, y)$ and for $j = 0, \dots, d//6$ with d the number of features, the positional encoding is given by:

$$w_j = 1/(\tau^j), \quad p_{j,i} = sin(i * w_j), cos(i * w_j) \tag{2}$$

Joint Training To learn from N datasets, we adopt a multi-task learning paradigm and equip the superformer $\mathcal{A}$ with N classification heads $\{\texttt{MLP}_i\}_{i=1}^{N}$. Given E epochs, $\mathcal{A}$ is pre-trained on ImageNet for $E/3$, and subsequently "fine-tuned" on all datasets, including ImageNet, for the remaining epochs. During fine-tuning, we sample at each optimizer step a different dataset according to a weighted task-sampling strategy with a probability p_i for task i proportional to the cardinality of the dataset and given by $p_i = \frac{\#\text{training steps}_i}{\text{tot. \# training steps}}$. We compute the gradient at each step and update the parameters. As proven in [27], co-training with a weighted task sampling has a regularization effect on smaller datasets, and mitigates over-fitting. The model is trained with binary cross-entropy loss, and with an additional term to measure the distribution gap between one-shot and stand-alone models. The optimization problem is:

$$\begin{aligned} \boldsymbol{w}^*_{\mathcal{A}}, o^*_{\mathcal{A}} = &\underset{\boldsymbol{w}_{\mathcal{A}}, o_{\mathcal{A}}}{\text{argmax}} \log p(D_i|\boldsymbol{w}_{\mathcal{A}}, \mathcal{A}, o_{\mathcal{A}}) \\ &\underbrace{+ \delta(o_{\alpha_i}, o_{\alpha_j}) \sum_{i,j=1}^{N} \log p(D_i|\boldsymbol{w}_{\alpha_i}, \boldsymbol{w}_{\alpha_j}, \alpha_i, \alpha_j)}_{\text{new term Distribution Gap (DG)}}, \end{aligned} \tag{3}$$

where D_i represents the dataset i, $\boldsymbol{w}_{\mathcal{A}}$ are the weights of the supernetwork, $\alpha_{i,j}$ are two different sub-networks, and the new term $o_{\mathcal{A}}$ expresses the optimization of the local weights assignment that provides a measurement for the extent to which two architectures can share weights. Although video datasets offer useful motion information, the frame redundancy and smaller dataset size compared to image datasets may limit the spatial content richness inside a video clip. Simultaneous training on multiple video datasets might limit a model's ability to learn appearance information effectively. While appearance information can be initially acquired through pre-training on image datasets, robust spatial representations may be diluted during fine-tuning on spatially redundant video data [27], [71] [5]. To address this, continuous training of object recognition with action recognition during fine-tuning could help maintain these representations and enhance model performance. Furthermore, action recognition feature extractors that are jointly learned generate more versatile video representations that can be applied to various datasets without additional fine-tuning. As previously noted [2], this approach emphasizes how crucial it is to increase both the number and scope of action recognition samples. Attention-based models might be prone to overfitting on narrower video distributions, diminishing the generalization of learned representations. Additionally, different datasets may emphasize distinct inductive biases in video modeling, such as temporal or spatial representation learning. More robust feature extractors that encode appearance and motion information may result from joint learning on both distributions, thereby enhancing performance on action recognition benchmarks.

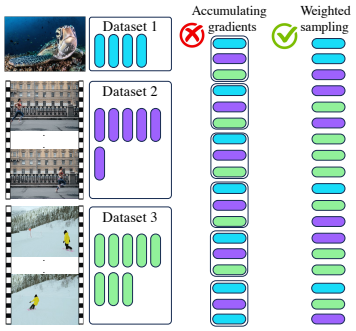

Figure 2. Dataset sampling schedule. The elements in a dataset represent the # of training steps performed. We sample each dataset *without* accumulating gradients, as we experienced to be detrimental for performance.

Evolution Given the trained supernet, we run a standard evolution search under resource constraints to obtain the optimal subnets with the objective of maximizing the accuracy while minimizing the model size. The evolution search starts by picking N random architectures as seeds. Parent networks are chosen among the top S sampled architectures to obtain "child models", and undergo crossover and mutation to obtain the next generation. In order to achieve crossover, two candidates are chosen at random and crossed to create a new candidate for each generation. Mutation, in the context of ViTs, consists in mutating a candidate depth with probability P_d and a block with a probability of P_m.

4. Experiments

We evaluate our approach on ImageNet [8], Kinetics 400, Kinetics 600 [22], and Something-Something-v2 [16]. These datasets were chosen to cover diverse challenges, as they enable the evaluation of our proposed approach's capability to find a unique architecture that can solve tasks belonging to two different modalities (ImageNet vs Kinetics) and tasks characterized by different inductive biases (spatial Kinetic vs temporal Something-Something). We use

Table 2. Search space definition. We set up 4 supernets that satisfy different resource constraints. The tuples of three values define the $\mathcal{C}$ dimensions of Eq. (1) and represent the lowest value, highest, and steps. Tubes are defined by temporal (k_1), spatial (k_2, k_3) dimensions, and by the offset o. Patch sizes are included in tube shapes, *e.g.* a 2D patch is obtained with a tube $1 \times 16 \times 16$, stride $s = (32, 16, 16)$ for a 32 frames video input.

SuperNet	Tiny	Small	Base	Large
Tube shapes		k_1 = (1, 32, 4), k_2 = (4, 32, 4), k_3 =(4, 32, 4) s_1 = (4, 32, 4), s_2 = (8, 32, 4), s_3 =(8, 32, 4) o_1 = (0, 8, 2), o_2 = (0, 16, 4), o_3 =(0, 16, 4)		
Embed Dim	(192, 240, 24)	(324, 444, 60)	(528, 624, 48)	(768, 90, 948)
Q-K-V Dim	(192, 256, 64)	(324, 444, 60)	(512, 640, 64)	(768, 90, 948)
MLP Ratio	(3.5, 4.0, 0.5)	(3.0, 4.0, 0.5)	(3.0, 4.0, 0.5)	(3.0, 4.0, 0.5)
Head Num	(3, 4, 1)	(5, 7, 1)	(8, 10, 1)	(13, 14, 1)
Depth Num	(12, 14, 1)	(14, 14, 1)	(14, 16, 1)	(20, 22, 1)
Params Range	5 - 10M	15 - 35M	44 - 76M	89 - 239M

Top-1 and Top-5 accuracies as evaluation metrics and report FLOPs and parameters of our approach and competitors, when available. Our main results are obtained following the joint-training strategy (Sec. 4.1). We conduct a series of ablation studies to validate the different components characterizing our approach (Sec. 4.2).

4.1. Main Results

We followed the joint-training strategy, sampling at each search step 4 tubes from the search space. We compare our approach to previous SOTA hand-designed and SOTA NAS-designed methods. Tab. 3, 4, 5, and 6 show the performance on ImageNet, Kinetics-400, Kinetics-600 and Something-Something-v2. NAS approaches mentioned in Sec. 2.2 not appearing in Tab. 4, 5 did not perform experiments on those datasets. These results show how our approach outperforms not only previous SOTA NAS methods on videos, which we expected as many [36, 60, 72] cannot be pre-trained on ImageNet due to their formulation but outperforms also hand-designed models of comparable size and in many cases of much bigger size, surpassing therefore architectures also in terms of efficiency. In Tab. 3, of particular notice, should be the comparison between our VIM-NAS-B and Tubes-ViT-B, where we gain more than 2 % in Top-1 accuracy with 30M less parameters, and between our VIM-NAS-B and DeepMAD, where we have comparable performance with less parameters. We also surpass by 0.1 % the current best NAS method[1] on ImageNet. Despite our model having approximately twice the number of parameters compared to EfficientNet-B7, it achieves faster inference and requires fewer GFLOPs. The good and consistent results on ImageNet are confirmed in our experiments with video datasets, where we experienced even larger improvements, particularly when compared to other NAS methods. Indeed, as Tab. 4 highlights, we surpass previous NAS methods of more than 9% in Top-1 accuracy, and the strongest hand-designed method (TubesViT-L) by 0.5 % using almost $1/3$ of the parameters. All the sizes of our model perform well, even compared to hand designed models that use significantly larger pre-training data (e.g., CoCa with 1B params and 1.8B examples, MerlotReserve has 644M params and uses YT-1B dataset). Similar outcomes are observed for Kinetics-600 (refer to Tab. 5), where current state-of-the-art TubesViT-H model is surpassed by 0.4 %. It is noteworthy that a direct comparison with previous Neural Architecture Search (NAS) methodologies for Video Action Recognition was not feasible. To the best of our knowledge, there is a lack of existing NAS approaches that have conducted experiments on the Kinetics-600 dataset. Consequently, we establish the first NAS result for this dataset.

Table 3. Performance on ImageNet-1k. Results are grouped in hand designed (1^{st} block), NAS designed (2^{nd}), and our models (3^{rd}). Colors highlight comparisons in terms of accuracy and parameters or accuracy and FLOPs. Bold highlights best results.

Method	Resolution	Params	FLOPs	Top-1	Top 5
DeiT-T [51]	224^2	5.7M	1.2G	72.2 %	91.1 %
ResNet-50 [18]	224^2	25.5M	4.1G	76.2 %	91.4 %
ViT-S/16 [11]	384^2	22.1M	4.7G	78.8 %	-
DeiT-S [51]	224^2	22.1M	4.6G	79.9 %	95.0 %
ViT-B/16 [11]	384^2	86M	18G	79.7 %	-
ConvNeXt-T [31]	224^2	28.6M	4.5G	82.5 %	96.1 %
BoTNet-S1-59 [44]	224^2	33.5M	7.3G	81.7 %	95.8 %
TubesViT-B [38]	224^2	86M	12G	81.4 %	95.2 %
DeepMAD [42]	224^2	89M	16G	84.0 %	-
Autoformer-T [62]	224^2	5.7M	1.3G	74.7 %	92.6 %
MobileNetV3 [19]	224^2	5.4M	0.22G	75.2 %	-
EfficientNet-B0 [49]	224^2	5.4M	0.39G	77.1 %	93.3 %
Autoformer-S [62]	224^2	22.9M	5.1G	81.7 %	95.7 %
ViTAS-Twins-S [45]	224^2	30.5M	3.0	82.0 %	95.7 %
Autoformer-B [62]	224^2	54M	11G	82.4 %	95.7 %
ViTAS-Twins-B [45]	224^2	66M	8.8G	83.5 %	96.5 %
EfficientNet-B7 [49]	600^2	66M	37G	84.3 %	97.0 %
VIM-NAS-T	224^2	5.5M	1.3G	79.3 %	93.0 %
VIM-NAS-S	224^2	22.9M	3.9G	81.9 %	96.0 %
VIM-NAS-B	224^2	53.7M	9.9G	83.9 %	96.6 %
VIM-NAS-L	224^2	128M	33G	**84.4%**	**97.3%**

[1] according to Papers with Code

Table 4. Kinetics-400. Results are grouped in hand designed (1^{st} block), NAS designed (2^{nd}), and our found models (3^{rd}).

Method	Frames	Params	FLOPs	Top-1	Top-5
I3D [4]	64	12.0M	108G	71.1%	90.3%
TSM R50 [28]	16	24.3M	65.0G	74.7%	-
2-Stream I3D [4]	64	25.0M	216G	75.7%	-
X3D-L [12]	16	-	24.8G	77.5%	92.9%
SlowFast 8×8 R101 [13]	8	53.7M	106G	77.9%	-
TimeSformer-L [2]	-	-	7.14T	80.7%	94.7%
ViViT-L FE [1]	-	-	11.94T	81.7%	93.8%
MAE-ST [14]	-	-	-	84.4%	-
VideoMAE [50]	16	632M	1.2T	87.4%	97.6%
TubesViT-B [38]	64	89M	0.87T	88.6%	97.6%
MVT-H [64]	32	-	73.57T	89.9%	98.3%
TubesViT-L [38]	64	311M	9.53T	90.2%	98.6%
TubesViT-H [38]	64	633M	13.2T	90.9%	98.6%
AutoX3D-S [20]	12	3.5M	2.9G	74.7%	-
AutoX3D-M [20]	16	3.5M	6.8G	76.7%	-
EvaNet [36]	-	-	-	77.2%	-
VAR with NAS [72]	16	-	15.78G	78.2%	93.1%
PV-NAS-L [60]	8	-	22.14G	78.7%	93.5%
AutoX3D-L [20]	16	6.2M	27.8G	78.8%	-
PV-NAS-M [60]	8	-	82.11G	81.4%	94.2%
VIM-NAS-T	32	5.5M	54.75G	83.1%	94.1%
VIM-NAS-S	32	22.9M	0.22T	88.9%	97.7%
VIM-NAS-B	32	53.7M	0.52T	90.5%	98.7%
VIM-NAS-L	32	128M	3.1T	**90.7%**	**98.8%**
VIM-NAS-H	32	412M	4.2T	**91.2%**	**99.1%**

Table 5. Kinetics-600 results: hand designed (1^{st} block), NAS designed (2^{nd}), and our found models (3^{rd}).

Method	Params	FLOPs	Top-1	Top-5
SlowFast R101-NL [13]	-	7.02T	81.8 %	95.1 %
X3D-L [12]	6.1M		81.9 %	95.5 %
TimeSformer-L [2]	-	7.14T	82.2 %	95.6 %
ViViT-L-FE [1]	-	11.94T	82.9 %	94.6 %
MViT-B [64]	36.8M	4.10T	83.8 %	96.3 %
ViViT-H [1]	-	47.77T	85.8 %	96.5 %
Florence [69]	647M	-	87.8 %	97.0 %
CoCa [67]	1B	-	89.4 %	-
MVT-H [65]	-	73.57T	90.3 %	98.5 %
TubesViT-B [38]	89M	0.87T	90.9 %	97.3 %
Merlot-Reserve-L [70]	-	-	91.1 %	97.1 %
TubesViT-L [38]	311M	9.53T	91.5 %	98.7 %
TubesViT-H [38]	635M	17.64T	91.8 %	98.9 %
	/			
VIM-NAS-S	22.9M	0.22T	91.3 %	98.2 %
VIM-NAS-B	53.7M	0.52T	91.7 %	99.0 %
VIM-NAS-L	128M	3.1T	**92.2 %**	**99.2 %**

Table 6. Something-Something-V2.

Method	Frames	Params	Top-1	Top-5
TSM [28]	16	24.3M	61.2 %	-
TimeSformer-B [2]	16	121.4M	62.5 %	-
X3D-M [12]	16	3.8M	62.7 %	-
VidTR-L [25]	32	-	63.0 %	-
CoVeR [71]	16	-	64.7 %	-
GSF [46]	16	24.1M	65.73 %	-
ViViT-L FE [1]	-	-	65.9 %	89.9 %
VoV3D-L [24]	32	5.8M	67.3 %	90.5 %
MFormer-L [34]	-	-	68.1 %	91.2 %
MViT [26]	40	213M	73.3 %	94.1 %
MaskFeat	40	218M	75.0 %	95.0 %
VideoMAE [50]	16	632M	75.4 %	95.2 %
Tubes-ViT-L [38]	32	311M	76.1 %	95.2 %
Auto-X3D-S [20]	13	3.5M	62.1 %	-
PV-NAS-L [60]	8	-	62.5 %	88.4 %
Auto-X3D-M [20]	16	5.3M	63.4 %	-
VIM-NAS-T	32	5.5M	**65.1 %**	**89.2 %**
VIM-NAS-S	32	22.9M	**71.1 %**	**93.8 %**
VIM-NAS-B	32	53.7M	**75.9 %**	**95.5 %**
VIM-NAS-L	32	128M	**76.9 %**	**96.3 %**
VIM-NAS-H	32	412M	**77.5 %**	**97.2 %**

Finally, we present the outcomes achieved for Something-Something-v2, which posed a distinctive challenge due to its substantial difference from other datasets used in our experiments. This dataset is characterized by a pronounced temporal component. As emphasized in Sec. 4.2, including the tube shapes in the search space is fundamental when addressing Something-Something-v2 (Tab. 9). Not doing so may result in a significant performance decline, likely attributed to the varying importance of the temporal component compared to other datasets.

4.2. Ablations

Subnet Performance without Retraining Following standard one-shot approaches, we ablate the effectiveness of the differentiable weight-entanglement mechanism with the additional local weight-assignment optimization by checking the performance of the subnetworks i) when directly inheriting the weights from the supernetwork, ii) when additional fine-tuning for 40 epochs, and iii) when retraining them from scratch for 300 epochs. Tab. 7 shows that if further finetuning or retraining the found subnets on ImageNet and Kinetics-400 lead to negligible performance gains, especially for ImageNet. However, we observe a more pronounced decrease (0.4 %) in performance for VIM-NAS-L on Kinetics-400, compared to other results, which we further examined in Tab. 8a. Finally, a significant portion of subnets perform exceptionally well when they inherit weights from the supernets. For VIM-NAS-T on Imagenet for example, all subnetworks consistently achieve Top-1 accuracies within the range of 77.8% to 79.3 %.

Benefits of joint-training As Tab. 8 highlights, joint-training brings us several benefits, in addition to the fact that the NAS procedure is $O(1)$ on the number of N datasets. Two crucial insights emerge from these experiments: i) joint-training yields a substantial boost in absolute Top-1

Table 7. Comparison of subnets with inherited weights, fine-tuned (40 epochs), trained from scratch (300 epochs) for ImageNet and Kinetics-400. Results for K400 were obtained by searching the architecture and tubes only on Kinetics.

	Model	Inherited	Finetune	Retrain
ImageNet	Tiny	79.3 %	79.4 %	79.4 %
	Small	81.9 %	82.0 %	81.9 %
	Base	83.4 %	83.4 %	83.4 %
K400	Small	83.0 %	83.2 %	83.1 %
	Base	84.4 %	84.5 %	84.6 %
	Large	84.5 %	84.9 %	84.9 %

accuracy not only for the subnets on video datasets but also for image datasets (Tab. 8b). We guess the gain in ImageNet performance is probably due to a regularization effect the other datasets perform on the transformer architecture that would otherwise overfit. ii) The previously mentioned drop for Kinetics-400 when compared to fine-tuned subnets and re-trained ones is not experienced anymore (Tab. 8a last row). Tab. 8b illustrates more in detail the advantages conferred by the joint-training process on the searching procedure. Our experiments include searching the optimal ViT and tubes directly on the target dataset (1^{st} column), searching the optimal architecture on ImageNet and fine-tuning the evolved subnet on the target dataset (2^{nd} column), and searching the optimal ViT and tubes with our joint-datasets setup. As expected, there is a drop if one single video dataset is directly deployed, leading however to results that still outperform previous NAS approaches. More surprisingly, we experienced a large gain when directly searching an architecture for ImageNet, and finetuning the topology (while training from scratch the 3D tubes) on the video datasets. We compare an ordinal weight-entanglement mechanism [62], a differentiable mechanism with weight-entanglement (without the new term of Eq. (3)) [48], a cyclic weight-entanglement [45], and a gradient-conflict aware method [15]. We chose as comparisons approaches that were proposed to reduce the gap between inherited subnetworks and trained supernetwork. Tab. 9a highlights that VIM-NAS achieves the highest Top-1 accuracy both on ImageNet and Kinetics-400, and proves more in general the benefits of our setup independently on the specific ViT searching procedure, which we hope will push other researchers to investigate and advance on NAS for video action recognition. If we compare these results with previous single-dataset NAS methods (Tab. 4), all lead to an improvement. Tab. 9b proves the benefits of the new loss term in Eq. (3) and of searching both the architecture and the shape of the tubes/patches.

5. Conclusion

We proposed a new one-shot architecture search method for Video and IMage Recognition (VIM-NAS). VIM-NAS exploits a differentiable weight-entanglement mechanism, and an additional optimization objective for local-weights assignment to search for optimal ViT architectures and optimal tube shapes. The tubes guarantee the same architecture to process both image and video input and the procedure to be run in $O(1)$ with respect to the number of datasets. Our joint-training strategy, combined with our optimization, improves previous NAS-SOTA performance on the video datasets. Differently from previous NAS methods for Video Action Recognition, we propose a framework that aims at closing the gap with NAS approaches designed for image tasks, and that can be compared to hand-designed models, as it exploits ImageNet pretraining. Our approach achieves impressive results also on the challenging Something-SomethingV2 dataset, rarely investigated by previous methods. Applying such an approach to convolutional networks search is a potential research direction.

Table 8. Benefits of joint-training.

(a) Effects of combining datasets on inherited weights. K400 (1^{st} block), K400+IMNET (2^{nd} block).

	Inherited	Finetune	Retrain
S	83.0 %	83.2 %	83.1 %
B	84.4 %	84.5 %	84.6 %
L	84.5 %	84.9 %	84.9 %
S	88.7 %	88.8 %	88.7 %
B	89.9 %	89.9 %	89.9 %
L	90.1 %	90.1 %	90.2 %

(b) Performance comparison when directly searching on the target dataset $\mathcal{D}_T$, when searching on ImageNet and fine-tuning (FT) the found ViT on the target dataset, and when searching with our join-modality.

	$\mathcal{D}_T$	IMNET + FT	Joint
IMNET (T)	78.0%	78.3% $\uparrow_{0.3\%}$	79.3% $\uparrow_{1.3\%}$
K400 (B)	84.4 %	89.3 % $\uparrow_{4.9\%}$	90.5 % $\uparrow_{6.1\%}$
K600 (B)	85.5 %	90.4 % $\uparrow_{4.9\%}$	91.7 % $\uparrow_{6.2\%}$
SSV2 (T)	67.2 %	68.0 % $\uparrow_{0.8\%}$	69.1 % $\uparrow_{1.9\%}$

Table 9. Ablation studies on the components of our method.

(a) Comparison with other ViT search methods under our setup on ImageNet and Kinetics-400. Model deployed: VIM-NAS-S.

	Autoformer [62]	ViTAS [45]	GraVIT-E [48]	NASViT [15]	Ours
IMNET	80.8%	80.9%	81.5%	81.0%	81.9%
K400	88.0%	88.5%	88.6%	88.1%	88.9%
Params	19.8M	24.8M	21.1M	22.2M	22.9M

(b) Ablation on VIM-NAS components on ImageNet, Kinetics-400, and Something-SomethingV2. Experiments run with ViT-Base and Superformer-Base, with joint training applied.

	No NAS	NAS on ViT	NAS on ViT&Tubes w/o DG Eq. (3)	NAS on ViT&Tubes w/ DG
IMNET	81.3%	82.0%	82.7 %	83.4%
K400	88.6%	88.8%	90.2 %	90.5%
SSV2	75.9%	76.0%	76.3 %	76.9%

Acknowledgement

This work has been partially supported by the project IN2814 of Free University of Bozen-Bolzano, by the Spanish project PID2022-136436NB-I00 and by ICREA under the ICREA Academia programme.

References

[1] Anurag Arnab, Mostafa Dehghani, Georg Heigold, Chen Sun, Mario Lucic, and Cordelia Schmid. Vivit: A video vision transformer. *2021 IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 6816–6826, 2021. 3, 7

[2] Gedas Bertasius, Heng Wang, and Lorenzo Torresani. Is space-time attention all you need for video understanding? In *Proceedings of the 38th International Conference on Machine Learning*, pages 813–824. PMLR, 2021. 3, 5, 7

[3] Han Cai, Chuang Gan, Tianzhe Wang, Zhekai Zhang, and Song Han. Once-for-all: Train one network and specialize it for efficient deployment. In *International Conference on Learning Representations*, 2020. 1

[4] João Carreira and Andrew Zisserman. Quo vadis, action recognition? a new model and the kinetics dataset. *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 4724–4733, 2017. 2, 3, 7

[5] Rich Caruana. Multitask learning. *Machine Learning*, 28: 41–75, 1997. 5

[6] Xiangning Chen, Ruochen Wang, Minhao Cheng, Xiaocheng Tang, and Cho-Jui Hsieh. Drnas: Dirichlet neural architecture search. *ArXiv*, abs/2006.10355, 2020. 1

[7] Xiaoliang Dai, Alvin Wan, Peizhao Zhang, Bichen Wu, Zijian He, Zhen Wei, Kan Chen, Yuandong Tian, Matthew Yu, Peter Vajda, and Joseph E. Gonzalez. Fbnetv3: Joint architecture-recipe search using predictor pretraining. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 16276–16285, 2021. 1

[8] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *2009 IEEE Conference on Computer Vision and Pattern Recognition*, pages 248–255, 2009. 5

[9] Jeff Donahue, Lisa Anne Hendricks, Marcus Rohrbach, Subhashini Venugopalan, Sergio Guadarrama, Kate Saenko, and Trevor Darrell. Long-term recurrent convolutional networks for visual recognition and description. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 39(4):677–691, 2017. 3

[10] Xuanyi Dong and Yi Yang. Nas-bench-201: Extending the scope of reproducible neural architecture search. *CoRR*, abs/2001.00326, 2020. 2

[11] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, Jakob Uszkoreit, and Neil Houlsby. An image is worth 16x16 words: Transformers for image recognition at scale. In *International Conference on Learning Representations*, 2021. 2, 5, 6

[12] Christoph Feichtenhofer. X3d: Expanding architectures for efficient video recognition. *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 200–210, 2020. 7

[13] Christoph Feichtenhofer, Haoqi Fan, Jitendra Malik, and Kaiming He. Slowfast networks for video recognition. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, 2019. 2, 7

[14] Christoph Feichtenhofer, Haoqi Fan, Yanghao Li, and Kaiming He. Masked autoencoders as spatiotemporal learners. In *Advances in Neural Information Processing Systems*, 2022. 7

[15] Chengyue Gong, Dilin Wang, Meng Li, Xinlei Chen, Zhicheng Yan, Yuandong Tian, qiang liu, and Vikas Chandra. NASVit: Neural architecture search for efficient vision transformers with gradient conflict aware supernet training. In *International Conference on Learning Representations*, 2022. 8

[16] Raghav Goyal, Samira Ebrahimi Kahou, Vincent Michalski, Joanna Materzyńska, Susanne Westphal, Heuna Kim, Valentin Haenel, Ingo Fruend, Peter Yianilos, Moritz Mueller-Freitag, Florian Hoppe, Christian Thurau, Ingo Bax, and Roland Memisevic. The "something something" video database for learning and evaluating visual common sense, 2017. 1, 5

[17] Zichao Guo, Xiangyu Zhang, Haoyuan Mu, Wen Heng, Zechun Liu, Yichen Wei, and Jian Sun. Single path one-shot neural architecture search with uniform sampling. In *Computer Vision – ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part XVI*, page 544–560, Berlin, Heidelberg, 2020. Springer-Verlag. 1

[18] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016. 6

[19] Andrew Howard, Mark Sandler, Bo Chen, Weijun Wang, Liang-Chieh Chen, Mingxing Tan, Grace Chu, Vijay Vasudevan, Yukun Zhu, Ruoming Pang, Hartwig Adam, and Quoc Le. Searching for mobilenetv3. In *2019 IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 1314–1324, 2019. 6

[20] Yifan Jiang, Xinyu Gong, Junru Wu, Humphrey Shi, Zhicheng Yan, and Zhangyang Wang. Auto-x3d: Ultra-efficient video understanding via finer-grained neural architecture search. In *2022 IEEE/CVF Winter Conference on Applications of Computer Vision (WACV)*, pages 2354–2363, 2022. 1, 2, 3, 7

[21] Andrej Karpathy, George Toderici, Sanketh Shetty, Thomas Leung, Rahul Sukthankar, and Li Fei-Fei. Large-scale video classification with convolutional neural networks. In *2014 IEEE Conference on Computer Vision and Pattern Recognition*, pages 1725–1732, 2014. 2

[22] 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, 2017. 5

[23] Hayeon Lee, Eunyoung Hyung, and Sung Ju Hwang. Rapid neural architecture search by learning to generate graphs from datasets. *European Conference on Computer Vision (ECCV)*, abs/2107.00860, 2021. 1

[24] Youngwan Lee, Hyung-Il Kim, Kimin Yun, and Jinyoung Moon. Diverse temporal aggregation and depthwise spatiotemporal factorization for efficient video classification. *IEEE Access*, 9:163054–163064, 2021. 7

[25] Xinyu Li, Yanyi Zhang, Chunhui Liu, Bing Shuai, Yi Zhu, Biagio Brattoli, Hao Chen, Ivan Marsic, and Joseph Tighe. Vidtr: Video transformer without convolutions. *2021 IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 13557–13567, 2021. 7

[26] Yanghao Li, Chao-Yuan Wu, Haoqi Fan, Karttikeya Mangalam, Bo Xiong, Jitendra Malik, and Christoph Feichtenhofer. Mvitv2: Improved multiscale vision transformers for classification and detection, 2022. 7

[27] Valerii Likhosherstov, Anurag Arnab, Krzysztof Choromanski, Mario Lucic, Yi Tay, Adrian Weller, and Mostafa Dehghani. Polyvit: Co-training vision transformers on images, videos and audio. *Trans. Mach. Learn. Res.*, 2023, 2021. 2, 5

[28] Ji Lin, Chuang Gan, and Song Han. Tsm: Temporal shift module for efficient video understanding. In *Proceedings of the IEEE International Conference on Computer Vision*, 2019. 2, 3, 7

[29] Hanxiao Liu, Karen Simonyan, Oriol Vinyals, Chrisantha Fernando, and Koray Kavukcuoglu. Hierarchical representations for efficient architecture search. In *6th International Conference on Learning Representations, ICLR 2018, Vancouver, BC, Canada, April 30 - May 3, 2018, Conference Track Proceedings*. OpenReview.net, 2018. 3

[30] Hanxiao Liu, Karen Simonyan, and Yiming Yang. Darts: Differentiable architecture search. In *International Conference on Learning Representations*, 2018. 1, 4

[31] Zhuang Liu, Hanzi Mao, Chao-Yuan Wu, Christoph Feichtenhofer, Trevor Darrell, and Saining Xie. A convnet for the 2020s, 2022. 6

[32] Joe Yue-Hei Ng, M. Hausknecht, S. Vijayanarasimhan, O. Vinyals, R. Monga, and G. Toderici. Beyond short snippets: Deep networks for video classification. In *2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 4694–4702, Los Alamitos, CA, USA, 2015. IEEE Computer Society. 3

[33] Junyi Pan, Chong Sun, Yizhou Zhou, Ying Zhang, and Chen Li. Distribution consistent neural architecture search. In *2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 10874–10883, 2022. 3

[34] Mandela Patrick, Dylan Campbell, Yuki Asano, Ishan Misra, Florian Metze, Christoph Feichtenhofer, Andrea Vedaldi, and Joao F. Henriques. Keeping your eye on the ball: Trajectory attention in video transformers. In *Advances in Neural Information Processing Systems*, 2021. 7

[35] Wei Peng, Xiaopeng Hong, and Guoying Zhao. Video action recognition via neural architecture searching. *2019 IEEE International Conference on Image Processing (ICIP)*, pages 11–15, 2019. 1, 3

[36] A. J. Piergiovanni, Anelia Angelova, Alexander Toshev, and Michael S. Ryoo. Evolving space-time neural architectures for videos. *2019 IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 1793–1802, 2018. 1, 2, 3, 6, 7

[37] A. J. Piergiovanni, Anelia Angelova, and Michael S. Ryoo. Tiny video networks. *ArXiv*, abs/1910.06961, 2019. 3

[38] A. J. Piergiovanni, Weicheng Kuo, and Anelia Angelova. Rethinking video vits: Sparse video tubes for joint image and video learning. *CoRR*, abs/2212.03229, 2022. 2, 3, 5, 6, 7

[39] Zhaofan Qiu, Ting Yao, and Tao Mei. Learning spatio-temporal representation with pseudo-3d residual networks. In *2017 IEEE International Conference on Computer Vision (ICCV)*, pages 5534–5542, 2017. 2

[40] Pengzhen Ren, Gang Xiao, Xiaojun Chang, Yun Xiao, Zhihui Li, and Xiaojiang Chen. Nas-tc: Neural architecture search on temporal convolutions for complex action recognition. *ArXiv*, abs/2104.01110, 2021. 1, 2, 3

[41] Michael S. Ryoo, A. J. Piergiovanni, Mingxing Tan, and Anelia Angelova. Assemblenet: Searching for multi-stream neural connectivity in video architectures. *ArXiv*, abs/1905.13209, 2019. 2, 3

[42] Xuan Shen, Yaohua Wang, Ming Lin, Yilun Huang, Hao Tang, Xiuyu Sun, and Yanzhi Wang. Deepmad: Mathematical architecture design for deep convolutional neural network. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2023. 6

[43] Karen Simonyan and Andrew Zisserman. Two-stream convolutional networks for action recognition in videos. In *Advances in Neural Information Processing Systems*. Curran Associates, Inc., 2014. 2

[44] Aravind Srinivas, Tsung-Yi Lin, Niki Parmar, Jonathon Shlens, Pieter Abbeel, and Ashish Vaswani. Bottleneck transformers for visual recognition, 2021. 6

[45] Xiu Su, Shan You, Jiyang Xie, Mingkai Zheng, Fei Wang, Chen Qian, Changshui Zhang, Xiaogang Wang, and Chang Xu. Vision transformer architecture search. In *European Conference on Computer Vision*, 2021. 2, 3, 6, 8

[46] Swathikiran Sudhakaran, Sergio Escalera, and Oswald Lanz. Gate-shift-fuse for video action recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 45: 10913–10928, 2022. 7

[47] Swathikiran Sudhakaran, Sergio Escalera, and Oswald Lanz. Gate-shift-fuse for video action recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 45(9): 10913–10928, 2023. 3

[48] Rhea Sanjay Sukthanker, Arjun Krishnakumar, Sharat Patil, and Frank Hutter. Gravit-e: Gradient-based vision transformer search with entangled weights. In *Sixth Workshop on Meta-Learning at the Conference on Neural Information Processing Systems*, 2022. 3, 4, 8

[49] Mingxing Tan and Quoc V. Le. Efficientnet: Rethinking model scaling for convolutional neural networks. *ArXiv*, abs/1905.11946, 2019. 6

[50] Zhan Tong, Yibing Song, Jue Wang, and Limin Wang. Videomae: Masked autoencoders are data-efficient learners for self-supervised video pre-training, 2022. 7

[51] Hugo Touvron, Matthieu Cord, Matthijs Douze, Francisco Massa, Alexandre Sablayrolles, and Herve Jegou. Training data-efficient image transformers & distillation through attention. In *Proceedings of the 38th International Conference on Machine Learning*, pages 10347–10357. PMLR, 2021. 6

[52] Du Tran, Lubomir Bourdev, Rob Fergus, Lorenzo Torresani, and Manohar Paluri. Learning spatiotemporal features with 3d convolutional networks, 2015. 2

[53] Du Tran, Heng Wang, Lorenzo Torresani, Jamie Ray, Yann LeCun, and Manohar Paluri. A closer look at spatiotemporal convolutions for action recognition. In *2018 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2018, Salt Lake City, UT, USA, June 18-22, 2018*, pages 6450–6459. Computer Vision Foundation / IEEE Computer Society, 2018. 3

[54] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Ł ukasz Kaiser, and Illia Polosukhin. Attention is all you need. In *Advances in Neural Information Processing Systems*. Curran Associates, Inc., 2017. 3

[55] Dilin Wang, Chengyue Gong, Meng Li, Qiang Liu, and Vikas Chandra. Alphanet: Improved training of supernets with alpha-divergence, 2021. 1

[56] Ruochen Wang, Minhao Cheng, Xiangning Chen, Xiaocheng Tang, and Cho-Jui Hsieh. Rethinking architecture selection in differentiable nas. In *International Conference on Learning Representation*, 2021. 4

[57] R. Wang, D. Chen, Z. Wu, Y. Chen, X. Dai, M. Liu, Y. Jiang, L. Zhou, and L. Yuan. Bevt: Bert pretraining of video transformers. In *2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 14713–14723, Los Alamitos, CA, USA, 2022. IEEE Computer Society. 3

[58] Shiguang Wang, Tao Xie, Jian Cheng, Xingcheng Zhang, and Haijun Liu. Mdl-nas: A joint multi-domain learning framework for vision transformer. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 20094–20104, 2023. 1, 2

[59] Xuanhan Wang, Lianli Gao, Peng Wang, Xiaoshuai Sun, and Xianglong Liu. Two-stream 3-d convnet fusion for action recognition in videos with arbitrary size and length. *IEEE Transactions on Multimedia*, 20(3):634–644, 2018. 3

[60] Zihao Wang, Chen Lin, Lu Sheng, Junjie Yan, and Jing Shao. Pv-nas: Practical neural architecture search for video recognition, 2020. 1, 3, 6, 7

[61] Wei Wen, Hanxiao Liu, Yiran Chen, Hai Li, Gabriel Bender, and Pieter-Jan Kindermans. Neural predictor for neural architecture search. In *Computer Vision – ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part XXIX*, page 660–676, Berlin, Heidelberg, 2020. Springer-Verlag. 1, 3

[62] Haixu Wu, Jiehui Xu, Jianmin Wang, and Mingsheng Long. Autoformer: Decomposition transformers with Auto-Correlation for long-term series forecasting. In *Advances in Neural Information Processing Systems*, 2021. 2, 3, 4, 6, 8

[63] Saining Xie, Chen Sun, Jonathan Huang, Zhuowen Tu, and Kevin P. Murphy. Rethinking spatiotemporal feature learning: Speed-accuracy trade-offs in video classification. In *European Conference on Computer Vision*, 2017. 2

[64] Shen Yan, Xuehan Xiong, Anurag Arnab, Zhichao Lu, Mi Zhang, Chen Sun, and Cordelia Schmid. Multiview transformers for video recognition. In *2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 3323–3333, 2022. 3, 7

[65] Shen Yan, Xuehan Xiong, Anurag Arnab, Zhichao Lu, Mi Zhang, Chen Sun, and Cordelia Schmid. Multiview transformers for video recognition. In *2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 3323–3333, 2022. 7

[66] Chris Ying, Aaron Klein, Esteban Real, Eric Christiansen, Kevin P. Murphy, and Frank Hutter. Nas-bench-101: Towards reproducible neural architecture search. In *International Conference on Machine Learning*, 2019. 2

[67] Jiahui Yu, Zirui Wang, Vijay Vasudevan, Legg Yeung, Mojtaba Seyedhosseini, and Yonghui Wu. Coca: Contrastive captioners are image-text foundation models. *Transactions on Machine Learning Research*, 2022. 7

[68] Zhou Yu, Yuhao Cui, Jun Yu, Meng Wang, Dacheng Tao, and Qi Tian. Deep multimodal neural architecture search. In *Proceedings of the 28th ACM International Conference on Multimedia*, page 3743–3752, New York, NY, USA, 2020. Association for Computing Machinery. 1

[69] Lu Yuan, Dongdong Chen, Yi-Ling Chen, Noel C. F. Codella, Xiyang Dai, Jianfeng Gao, Houdong Hu, Xuedong Huang, Boxin Li, Chunyuan Li, Ce Liu, Mengchen Liu, Zicheng Liu, Yumao Lu, Yu Shi, Lijuan Wang, Jianfeng Wang, Bin Xiao, Zhen Xiao, Jianwei Yang, Michael Zeng, Luowei Zhou, and Pengchuan Zhang. Florence: A new foundation model for computer vision. *ArXiv*, abs/2111.11432, 2021. 7

[70] Rowan Zellers, Jiasen Lu, Ximing Lu, Youngjae Yu, Yanpeng Zhao, Mohammadreza Salehi, Aditya Kusupati, Jack Hessel, Ali Farhadi, and Yejin Choi. Merlot reserve: Multimodal neural script knowledge through vision and language and sound. In *CVPR*, 2022. 7

[71] 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, 2021. 2, 5, 7

[72] Yuanding Zhou, Baopu Li, Zhihui Wang, and Haojie Li. Video action recognition with neural architecture search. In *Proceedings of The 13th Asian Conference on Machine Learning*, pages 1675–1690. PMLR, 2021. 1, 3, 6, 7

[73] Barret Zoph, Vijay Vasudevan, Jonathon Shlens, and Quoc V. Le. Learning transferable architectures for scalable image recognition, 2018. 3