# MDL-NAS: A Joint Multi-domain Learning Framework for Vision Transformer

Shiguang Wang[1,3] Tao Xie[2,3] Jian Cheng[1,*] Xingcheng Zhang[3] Haijun Liu[4]
[1]University of Electronic Science and Technology of China [2]Harbin Institute of Technology
[3]SenseTime Research [4]Chongqing University
xiaohu_wyyx@163.com xietao1997@hit.edu.cn
chenjian@uestc.edu.cn zhangxingcheng@sensetime.com haijun_liu@126.com

## Abstract

*In this work, we introduce MDL-NAS, a unified framework that integrates multiple vision tasks into a manageable supernet and optimizes these tasks collectively under diverse dataset domains. MDL-NAS is storage-efficient since multiple models with a majority of shared parameters can be deposited into a single one. Technically, MDL-NAS constructs a coarse-to-fine search space, where the coarse search space offers various optimal architectures for different tasks while the fine search space provides fine-grained parameter sharing to tackle the inherent obstacles of multi-domain learning. In the fine search space, we suggest two parameter sharing policies, i.e., sequential sharing policy and mask sharing policy. Compared with previous works, such two sharing policies allow for the partial sharing and non-sharing of parameters at each layer of the network, hence attaining real fine-grained parameter sharing. Finally, we present a joint-subnet search algorithm that finds the optimal architecture and sharing parameters for each task within total resource constraints, challenging the traditional practice that downstream vision tasks are typically equipped with backbone networks designed for image classification. Experimentally, we demonstrate that MDL-NAS families fitted with non-hierarchical or hierarchical transformers deliver competitive performance for all tasks compared with state-of-the-art methods while maintaining efficient storage deployment and computation. We also demonstrate that MDL-NAS allows incremental learning and evades catastrophic forgetting when generalizing to a new task.*

## 1. Introduction

Recently, transformers have become the standard pattern for natural language processing (NLP) tasks due to their efficacy in modelling long-range relationships via the self-attention mechanism [41]. Such success and good properties of transformers have spawned a slew of subsequent works that apply them to a wide variety of computer vision tasks, such as image classification [6,27,32,49], object detection [5,57], semantic segmentation [55], and video understanding [56], achieving impressive results. However, these methods only apply transformer to a specific domain. After observing the success of transformers, a naive question arises: could a transformer simultaneously handle multiple vision tasks under a variety of dataset domains ?

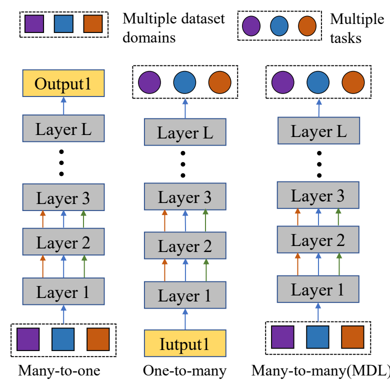
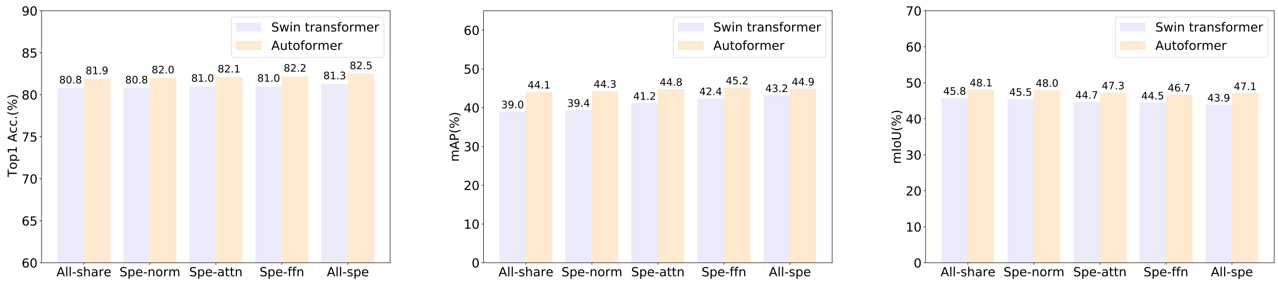



Figure 1. Illustration of differences between multi-domain learning (MDL) and other learning paradigms. MDL-NAS jointly optimizes multiple vision tasks under different dataset domains. L denotes the layer numbers in the backbone.

While a few works have investigated the usage of transformers to handle multiple input modalities (i.e., images and text), they typically concentrate on a particular task, such as visual question answering [20, 24], i.e., many-to-one mapping. Also, some methods [2,26] explore to simultaneously performing depth estimation, surface normal estimation, and semantic segmentation on a given input image. However, these methods are restricted to a single-domain setting, where all inputs are the same, i,e,, one-to-many mapping. In addition, there are some works [19, 28] that employ transformers to solve different tasks under multiple domains (multi-domain learning), which is more realistic, i.e., many-to-many mapping, as shown in Fig. 1. Never-

*Corresponding author.




(a) Image classification (b) Object detection (c) Semantic segmentation

Figure 2. Adjust baselines that shares all parameters in backbone with task-specific normalization (Spe-norm), task-specific self-attention (Spe-attn), task-specific feed forward network (Spe-ffn) and all task-specific parameters (All-spe) under the same training recipe.

theless, these methods utilize diverse encoders to manage different dataset domains, which is inefficient in terms of storage deployment. In this work, we investigate a unified network that optimizes multiple vision tasks over multiple dataset domains to enable all tasks to share as many parameters as feasible while maintaining promising performance. As a preliminary step, we conduct experiments to observe the performance impact of treating various components of vision transformers as task-specific parameters.

As depicted in Fig. 2, considering the multihead self-attention (MHSA) layer or feed-forward network (FFN) or LayerNorm (LN) throughout the backbone as task-specific parameters can all achieve a certain performance gain for classification and detection tasks over a baseline that shares all parameters. Besides, we observe that the performance of semantic segmentation is elevated when all parameters are shared, indicating that closely-related tasks have mutual benefits whereas some tasks have conflicts against each other under multi-domain learning setting. Consequently, to use task-shared parameters for learning task-reciprocal features while using task-specific parameters for mitigating conflicts, sharing parameters with various proportions inside each layer is an immediate thought, which motivates us to find a method to supply different share ratios for different layers in the network. Moreover, when optimizing multiple tasks collectively, we typically equip these tasks with the backbone designed for image classification, which may be sub-optimal due to the gap between the image classification task and other vision tasks.

To tackle these issues, we introduce MDL-NAS, a unified framework based on vision transformers, which accommodates multiple vision tasks under heterogeneous dataset domains into a modest supernet and jointly optimizes these tasks. Specifically, we first construct a coarse search space comprising embedding dimension, heads number, query/key/value dimension, and MLP ratios for each transformer block to discover different optimal architectures for diverse tasks. Moreover, such space comprises candidate architectures with a wide spectrum of model size, which provides certain flexibility for final model deployment. Based on the coarse search space, we design a fine search space that offers fine-grained parameter sharing for all tasks to resolve the inherent challenges of multi-domain learning. In the fine search space, we suggest two parameter sharing policies, namely sequential sharing policy and mask sharing policy. Sequential sharing policy enables all tasks to share parameters for each layer in order, which allows to customize the parameter share ratio. Mask sharing policy provides maximum flexibility for different tasks to share parameters with various proportions and channels inside each layer. Following Autoformer [6], to address the efficiency issue, we leverage the weight entanglement training strategy to train MDL-NAS, allowing thousands of subnets to be extremely well-trained.

During the search stage, we propose a joint-subnet search algorithm that finds the optimal architecture and sharing parameters for each task under total resource constraints. The searched subnets with various architectures share as many parameters as possible in the backbone, guaranteeing excellent performance for each task while keeping storage-efficient for model deployment.

Experiments show that the searched models with weights inherited from the supernet outperform several baselines and are comparable with the state-of-the-art methods that are trained individually for specific tasks. We also demonstrate that MDL-NAS allows incremental learning and evades catastrophic forgetting when generalizing to a new task. Thus, MDL-NAS is more parameter-efficient and can scale up more gracefully with the number of tasks increasing, as illustrated in Sec. 4.4.

The key contributions of this work can be summarized as: (1) We propose MDL-NAS that accepts multiple dataset domains as input to optimize multiple vision tasks concurrently. (2) We construct a coarse-to-fine search space, with the coarse search space finding optimal architectures for all tasks and the fine search space coupled with sequential or mask sharing policy providing fine-grained shared parameters to learn task-reciprocal features and extra task-specific parameters for learning task-related features. (3) We introduce a subnet search algorithm to jointly search architectures and share ratios, enabling all tasks to share as many parameters as feasible while ensuring high performance for each task. (4) We demonstrate that MDL-NAS allows incremental learning with fewer parameters.

## 2. Related Work

**Transformer in Vision.** Currently, numerous computer vision methods are actively applying the transformer to vision tasks. With the vision transformer (ViT) [13] as a starting point, various variants of vision transformers have lately been proposed to resolve the inherent challenges of ViT, such as data-efficient training [40], position embedding [11], effective tokenization [16, 54], multi-scale processing [10, 48] and hierarchical design [27, 44]. Note that for hierarchical design, Swin Transformer [27] and PVT [44] employ a pyramid structure like CNNs that downsamples the feature maps gradually, which is advantageous for downstream tasks, e.g., object detection and semantic segmentation. In this work, we do not propose any variant of vision transformer but investigate the utilization of both hierarchical and non-hierarchical vision transformers for simultaneously optimizing numerous vision tasks.

**Multi-task and multi-domain learning.** Multi-task [3, 12, 31, 35] and multi-domain [1, 4, 34, 45] learning have been enhanced dramatically as deep neural networks have become the de facto standard in computer vision frameworks. However, optimally utilizing their benefits remains a formidable challenges due to the effect of task conflicts or domain conflicts, i.e., gradient conflicts. Recent works tackle such conflicts by homogenizing gradients or architecture design. For homogenizing gradients, previous methods have narrowed the problem down to two types of differences (i.e., gradient magnitudes and directions) between task gradients and proposed several algorithms [8, 9, 22, 25, 36, 53] to homogenize these differences. For architecture design, Cross-Stitch Networks [30] contain one standard feed-forward network per task, with cross-stitch units to enable sharing of features among tasks. UberNet [23] proposes an image pyramid approach to process images across multiple resolutions, where for each resolution, additional task-specific layers are formed on the top of the shared VGG-Net [37]. However, these methods require a large number of network parameters and determine whether parameters are shared or not subjectively.

**Neural architecture search for transformer and multi-task learning.** Recently, researchers have leveraged supernet-based neural architecture search to find the optimal architecture for transformers. HAT [16] employs supernet for hardware-aware transformer optimization, which focuses mostly on NLP workloads. AutoFormer [6], ViTAS [38], and S3 [7] follow the central theme of CNN-based NAS methods [15, 52], leveraging NAS to optimize the ViT architecture, where the searched architectures achieve better accuracy than the naive vision transformer. When it comes to multi-task learning, AdaShare [39] adaptively decides what layers to share by using an efficient approach that jointly optimizes the network weights and policy distribution parameters. MTL-NAS [14] disentangles multi-task learning into task-specific backbones and general inter-task feature fusion connections. Compared to previous works [39, 43] (many-to-one or one-to-many mapping), we seek a unified framework that optimizes multiple mainstream vision tasks (classification, detection, and segmentation) under different dataset domains (many-to-many mapping), which is more challenge and realistic.

## 3. Method

### 3.1. Preliminary

This section begins with a brief review of the vision transformer (ViT) and Swin Transformer, which are representative examples of non-hierarchical and hierarchical vision transformers, respectively. ViT and Swin Transformer are also served as basic architectures of MDL-NAS.

**Vision transformer (ViT).** Given an input image $X \in \mathbb{R}^{H\times W\times C}$, ViT first reshapes it into a sequence of flattened 2D patches $X_P \in \mathbb{R}^{N\times(P^2\cdot C)}$ such that $C$ is the number of channel, ($H$, $W$) represents the resolution of the image and ($P$, $P$) is the resolution of each image patch. Thus, the sequence length $N$ is given by $N = HW/P^2$. ViT then leverages a trainable linear projection $W_{proj}$ to transform the patches to $D$ dimension vectors, i.e., patch embedding, where $D$ is called **embedding dimension**. A learnable [class] embedding $x_{cls} \in \mathbb{R}^D$ is injected into the front of the sequence of patch embeddings to serve as the image representation and 1D positional embeddings $E \in \mathbb{R}^{(N+1)\times D}$ are additionally added to the patch embeddings to keep positional information. Mathematically, the successive representation of the input sequence can be expressed as

$$X_0 = [x_{cls}, X_P W_{proj}] + E \tag{1}$$

where $W_{proj} \in \mathbb{R}^{(P^2\cdot C)\times D}$ is the linear projection parameter. The resultant sequence of embeddings is then fed into the transformer encoder, which is composed of alternating transformer blocks. Each transformer block consists of a multihead attention layer, a feed-forward neural network, residual connection, and layer normalization.

*Multihead Self-Attention (MHSA).* In the self-attention layer, the input sequence $X_0 \in \mathbb{R}^{(N+1)\times D}$ is first mapped into three different vectors: the query vector $Q \in \mathbb{R}^{N\times D_h}$, the key vector $K \in \mathbb{R}^{N\times D_h}$ and value vector $V \in \mathbb{R}^{N\times D_h}$, where $N$ is the number of tokens, $D$ is the embedding dimension, $D_h$ is the **Q-K-V dimension**. Subsequently, the attention function between different vectors is given by

$$Attention(Q, K, V) = softmax(\frac{QK^T}{\sqrt{d_h}})V, \tag{2}$$

where $\frac{1}{\sqrt{d_h}}$ is the scaling factor to boost gradient stability for improved training. Lastly, a fully connected layer is applied to project the dimension $D_h$ to $D$.

Multihead self-attention (MHSA) divides the query, key, and value vectors into different **heads number**, executes self-attention in parallel, and then projects their concatenated outputs. A residual connection is added to each multihead self-attention to strengthen the flow of information, followed by a layer normalization. The output of these operations can be described as:

$$Out_0 = LayerNorm(X_0 + MSHA(X_0)), \quad (3)$$

*Feed-Forward Network.* A feed-forward network is deployed after the multihead self-attention layer. It comprises two fully-connected layers and a nonlinear activation (e.g., GELU) function within them. A **MLP ratio** is applied between the two fully-connected layers.

**Swin Transformer.** Swin Transformer, an advanced ViT, seeks to incorporate several important visual priors, such as hierarchy, locality, and translation invariance, into the standard Transformer encoder, thereby combining the strengths of both: the basic Transformer unit has strong modelling capabilities while the visual priors make it advantageous for a variety of visual tasks. Methodologically, Swin Transformer handles input images in a hierarchical manner by employing multihead self-attention within non-overlapping local windows. It consists of four sequential stages with progressively diminishing input resolution and increasing **embedding dimension**, where each stage has **different number of transformer blocks** with the same embedding dimension. Since the non-overlapping partition strategy lacks cross-window connectivity, Swin Transformer recommends implementing shifted-window operations between each pair of succeeding window-based MHSA layers to stimulate cross-window interactions.

### 3.2. MDL-NAS

In this part, we present the proposed MDL-NAS that jointly learns numerous vision tasks across dataset domains with a moderate supernet. Specifically, MDL-NAS introduces a search space with varying granularity degrees, from coarse to fine. The coarse search space provides different optimal architectures for diverse tasks, whereas the fine search space provides fine-grained parameter sharing to tackle the inherent inadequacies of multi-task learning, e.g., shared parameter competition among tasks. We assume that MDL-NAS jointly optimizes $K$ tasks.

**Coarse search space $A_c$.** For non-hierarchical vision transformer (ViT), following AutoFormer [6], we search for several variable factors in each transformer building block, including embedding dimension, Q-K-V dimension, heads number, and MLP ratios, which are all critical for model capacity and performance. For instance, recent research [29] has demonstrated that utilizing a vast number of heads is not mandatory, even though it makes sense for each head to represent a unique representation subspace. As a result, we make the number of attention heads adaptable, enabling each attention module to find its own ideal number. Note that we fix the ratio $d_h$ of the Q-K-V dimension to the number of heads in each transformer block, making the scaling factor $\frac{1}{\sqrt{d_h}}$ in attention calculation invariant to the number of heads, hence enhancing gradient stability.

Since ViT employs constant widths throughout all of its blocks, we only need to search a single embedding dimension across the entirety of the models. For Swin Transformer, there are four successive stages with progressively decreasing input resolution and increasing embedding dimension, where each stage includes a different number of blocks with the same embedding dimension. Therefore, each stage has two search dimensions: number of blocks and embedding dimension. Each block in the stage contains a window-based multihead self-attention (MHSA) module and a feed-forward network (FFN) module. Following [7], We do not force the blocks in one stage to be identical. Thus, each block in the stage has several search dimensions including heads number, MLP ratio, Q-K-V embedding dimension.

**Fine search space $A_f$.** In contrast to previous works that would only allow different tasks either sharing or monopolizing all parameters in one layer, we investigate to share part parameters and monopolize others in a single layer, thereby enabling fine-grained parameter sharing. Thus, we design a fine search space that provides fine-grained share ratio in each layer of the transformers, concluding normalization layer and linear layer, etc. Moreover, in the fine search space, we suggest two sharing policies, namely sequential sharing policy and mask sharing policy.

Taking the $i$-th linear layer as an example, we define three search spaces for the layer, including input channel $C_{in}$, output channel $C_{out}$, and share ratio $\Lambda$. Accordingly, we initialize the weight of the layer as $W_i \in \mathbb{R}^{C_{out}^{max} \times C_{in}^{max}}$, where $C_{out}^{max}$ and $C_{in}^{max}$ are the maximum number of $C_{out}$ and $C_{in}$ respectively. Using a single weight, however, does not permit different tasks to share different ratios of such weight in various iterations. All tasks can only share $\tau$ ratio of the weight after training, where $\tau$ is the minimum number of $\Lambda$. To tackle this issue, we predefine another weight for the $i$-th layer as task-specific parameters $W_i^{spe}$, and view $W_i$ as task-shared parameters. Nextly, we illustrate the proposed two policies.

*Sequential sharing policy.* We predefine an all zeros vector $M_i = [0, 0, ..., 0] \in \mathbb{R}^{C_{out}^{max}}$ for the $i$-th layer to judge whether the channel in the layer is shared or not. For each batch in supernet training, we sample a number $c_{out}$ from $C_{out}$, $c_{in}$ from $C_{in}$, $\epsilon$ from $\Lambda$. According to sampled $c_{out}$ and $\epsilon$, we set the first $\epsilon \cdot c_{out}$ component of $M_i$ to 1 to derive

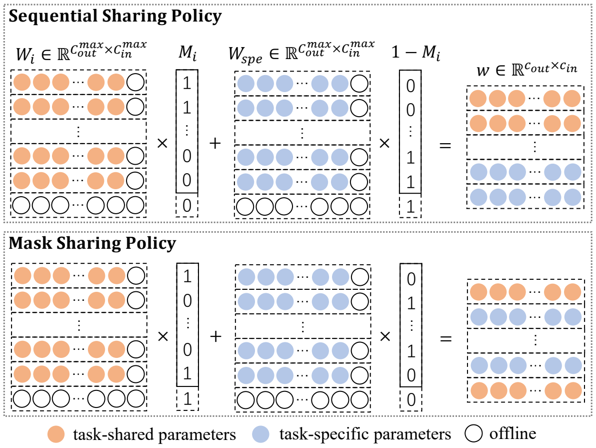



Figure 3. The illustration of dealing with the coarse and fine search space in a single layer with sequential sharing policy and mask sharing policy.

the shared mask as follows

$$M_i[: \epsilon \cdot c_{out}] = 1. \tag{4}$$

Then, we slice out the task-share weight $w_i^{share}$ for current batch by

$$w_i^{share} = W_i[: c_{out}, : c_{in}] \otimes M_i[: c_{out}], \tag{5}$$

and slice out the task-specific weight for current batch by

$$w_{k,i}^{spe} = W_i^{spe}[: c_{out}, : c_{in}] \otimes (1 - M_i)[: c_{out}], \tag{6}$$

where $\otimes$ denotes element-wise multiplication, as shown in Fig. 3; $k$ denotes the task index, $k = 1, 2, ..., K$.

Finally, we can obtain the current weight as $w_{k,i}$ by

$$w_{k,i} = w_i^{share} + w_{k,i}^{spe}, \tag{7}$$

where $w_{k,i}$ is used to produce the output for current batch for task $k$. Note that we omit the bias term in above process for simplicity.

*Mask sharing policy.* The sequential sharing policy can only enable multiple tasks to share the first partial channels in a certain layer in order, but must monopolize the remaining channels, sacrificing a certain degree of flexibility. To compensate, we propose mask sharing policy, a parameter-adaptive policy that permits diverse tasks to share varying quantities of parameters and channels in each layer with maximum flexibility.

For the $i$-th linear layer, we introduce scoring parameters $S_i = [S_i^j]$ for those channels in the layer where $j = 1, 2, ..., C_{out}^{max}$, and define the indicator function $\mathbb{I}(\cdot)$ as follows:

$$\mathbb{I}(S_i^j) = \begin{cases} 1, & if\ S_i^j \geq TH_r \\ 0, & otherwise \end{cases} \tag{8}$$

where $TH_r$ is a threshold. When $S_i^j \geq TH_r$, the $j$-th channel is transformed to a task-share parameter, and consequently optimize such parameters in the global group. Then, we can obtain the parameter sharing mask $M_i \in \mathbb{R}^{C_{out}^{max}}$ as follows

$$M_i = [\mathbb{I}(S_i^1), \mathbb{I}(S_i^2), ..., \mathbb{I}(S_i^{C_{out}^{max}})]. \tag{9}$$

Similarly, by sampling a number $c_{out}$ from $C_{out}$, $c_{in}$ from $C_{in}$, and $\epsilon$ from $\Lambda$, we derive the task-share weight $w_i^{share}$, $w_{k,i}^{spe}$ and $w_{k,i}$ according Eq. (5), Eq. (6), and Eq. (7), respectively. It is worth noting that we use $\epsilon$ as the threshold $TH_r$ for each iteration during training. The underlying key insight is that each layer' shared ratio is roughly bounded as a whole, while the learnable parameter $S_i$ is used to judge whether each channel inside each layer is shared or monopolized for the tasks. Since the gradient of indicator function $\mathbb{I}(\cdot)$ in Eq. (8) is zero at almost all points, we need to modify its gradient during backward pass, which is detailed in Appendix A.

The above two policies can be applied to all operations in each transformer block, including multihead self-attention layer, feed-forward network, and normalization layer. Besides, concluding additional task-specific parameters $W_i^{spe}$ into each layer does not result in a large increase in memory cost since only the sliced parameters of $W_i$ and $W_i^{spe}$ are updated at each iteration and all other parameters are kept offline. Moreover, in the inference and model deployment phase, we can slice out the parameters of $W_i$ and corresponding $W_i^{spe}$ for all tasks, and inject weights of all tasks into one to achieve efficient storage deployment.

### 3.3. Joint-subnet Search Algorithm

In this part, we introduce how to select optimal dedicated models from supernet for all tasks. Our goal is to find optimal architecture $\alpha_k$ of $A = \{A_c, A_f\}$ under resource constraints while maximize the overall performance $S\text{-}Score$ for all tasks, that is:

$$\begin{aligned} S\text{-}Score &= \max \sum_{k=1}^{K} \lambda_k \cdot f_k(\alpha_k), \\ s.t.\ \alpha_k \in A,\ & P^{share} + \sum_{k=1}^{K} P_k^{spe} \leq C \end{aligned} \tag{10}$$

where $k$ is the task index $k = 1, 2, ...K$; $\{\alpha_k | k = 1, 2, ...K\}$ have different architecture and share ratios inside each layer; $f_k(\alpha_k)$ is the performance of architecture $\alpha_k$ on task $k$; $\lambda_k$ is the weight coefficient of task $k$, which is set to 1 by default, and can be flexibly adjusted according to different tasks; $A$ is the whole search space of the supernet; $C$ is the total resource constraint (model size in this work); $P^{share}$ is the shared parameters across tasks and $P_k^{spe}$ is the specific parameters of task $k$, which are defined as follows:

$$P_{share} = \sum_{i=1}^{L} w_i^{share}, P_k^{spe} = \sum_{i=1}^{L} w_{k,i}^{spe}, \quad (11)$$

where $L$ is the layer numbers throughout the network.

Specifically, subnets for each task are evaluated and picked individually according to the manager of the evolution algorithm. Our objective here is to maximize $\sum_{k=1}^{K} \lambda_k \cdot f_k(\alpha_k)$ while minimizing the total constraints $C$. At the beginning of the evolution search, we pick $N$ random architectures as seeds for each task, and the top $J$ architectures are picked as parents for the next generation, which is generated through crossover and mutation. For crossover, two candidates are chosen at random and crossed to produce a new individual within each generation. For mutation, a candidate mutates its depth and each block with a probability of $P_m$ to produce a new architecture. Thanks to the designed coarse-to-fine search space, our search algorithm is capable of finding the ideal architecture for each task to ensure excellent performance while allowing as many parameters as possible to be shared among tasks to maximize storage efficiency.

# 4. Experiment

## 4.1. Implementation Details

**Searching space.** For non-hierarchical vision transformer (ViT), we design the seach space that includes: embedding dimension, heads num, Q-K-V dimension, MLP ratio, and share ratio. For hierarchical vision transformer (Swin Transformer), we also search the number of blocks in each stage.

**NAS pipeline on MDL-NAS.** MDL-NAS consists of three steps: supernet pre-training on ImageNet, joint-supernet fine-tuning on multiple dataset domains (ImageNet-1K, COCO, and ADE20K), and joint-subnet search with proposed algorithm for all tasks on the trained supernet.

The detailed search space and the implementation details of whole NAS pipeline are detailed in Appendix B and Appendix C, D and E, respectively.

## 4.2. Main Results

In this part, we compare MDL-NAS against baselines and the state-of-the-art methods on ImageNet-1K, COCO, and ADE20K datasets. We denote MDL-NAS that is built upon non-hierarchical (AutoFormer-B [6]) and hierarchical vision transformer (Swin-T [27]) as MDL-NAS-B$^\dagger$ and MDL-NAS-T$^\ddagger$, respectively. "+mask/seq" denotes that mask/sequential sharing policy is applied. The results are shown in Tab. 1, Tab. 2, Tab. 3, and Tab. 4. Notably, all MDL-NAS families inherit weights directly from supernets, without any retraining or postprocessing.

| Method | #Params | FLOPs | Top1 Acc. |
|---|---|---|---|
| ConvNets | | | |
| ResNet-34 [18] | 25.6M | 4.1G | 79.3 |
| ResNet-152 [18] | 60M | 11G | 80.6 |
| RegNetY-4G [33] | 21M | 4.0G | 80.0 |
| RegNetY-8G [33] | 39M | 8.0G | 81.7 |
| Vision Transformers | | | |
| DeiT-S [40] | 22M | 4.7G | 79.9 |
| DeiT-B [40] | 86M | 17.5G | 81.8 |
| PVT-S [44] | 25M | 3.8G | 79.8 |
| PVT-L [44] | 61M | 10G | 81.7 |
| T2T-ViT-24 [54] | 64M | 15G | 82.2 |
| Swin-T [27] | 29M | 4.5G | 81.3 |
| AutoFormer-B [6] | 54M | 11G | 82.4 |
| **MDL-NAS-B$^\dagger$ + mask** | **50M** | **10.8G** | **82.6** |
| **MDL-NAS-B$^\dagger$ + seq** | **55M** | **11.7G** | **82.9** |
| **MDL-NAS-T$^\ddagger$ + mask** | **27M** | **5.2G** | **81.7** |
| **MDL-NAS-T$^\ddagger$ + seq** | **28M** | **5.3G** | **81.7** |

Table 1. Comparison of MDL-NAS and previous methods on the ImageNet-1K validation set. Performances are measured with a single 224×224 crop. #Params refers to the number of parameters. FLOPs is calculated under the input scale of 224×224.

| Method | #Params | FLOPs | $AP^b$ | $AP^b_{50}$ | $AP^b_{75}$ | $AP^m$ | $AP^m_{50}$ | $AP^m_{75}$ |
|---|---|---|---|---|---|---|---|---|
| ResNet50 [18] | 44M | 260G | 41.0 | 61.7 | 44.9 | 37.1 | 58.4 | 40.1 |
| ResNet101 [18] | 63M | 336G | 42.8 | 63.2 | 47.1 | 38.5 | 60.1 | 41.3 |
| Xt101-64×4d [47] | 101M | 493G | 42.8 | 63.8 | 47.3 | 38.4 | 60.6 | 41.3 |
| PVT-S [44] | 44M | 245G | 43.0 | 65.3 | 46.9 | 39.9 | 62.5 | 42.8 |
| PVT-L [44] | 81M | 364G | 44.5 | 66.0 | 48.3 | 40.7 | 63.4 | 43.7 |
| Swin-T [27] | 48M | 264G | 46.0 | 68.2 | 50.2 | 41.6 | 65.1 | 44.8 |
| Focal-T [51] | 49M | 291G | 47.2 | 69.4 | 51.9 | 42.7 | 66.5 | 45.9 |
| Shuffle-T [21] | 48M | 268G | 46.8 | 68.9 | 51.5 | 42.3 | 66.0 | 45.6 |
| AutoFormer-B* [6] | 108M | 710G | 47.3 | 68.9 | 51.4 | 41.6 | 65.8 | 44.2 |
| **MDL-NAS-B$^\dagger$ + mask** | **116M** | **754G** | **48.2** | **69.7** | **52.6** | **42.2** | **66.3** | **45.0** |
| **MDL-NAS-B$^\dagger$ + seq** | **129M** | **813G** | **48.0** | **69.2** | **52.9** | **42.0** | **65.8** | **45.0** |
| **MDL-NAS-T$^\ddagger$ + mask** | **60M** | **309G** | **44.9** | **68.2** | **49.0** | **41.1** | **64.9** | **44.1** |
| **MDL-NAS-T$^\ddagger$ + seq** | **58M** | **299G** | **46.4** | **68.9** | **51.0** | **41.9** | **65.8** | **44.9** |

Table 2. COCO detection and segmentation with the Mask R-CNN. The performances are reported on the COCO val split under 3× schedules. The FLOPs (G) are measured at resolution 800 × 1280, and all models are pre-trained on the ImageNet-1K. In the table, * means our implementation that uses the searched backbone of Autoformer for object detection.

**ImageNet-1K classification.** As reported in Tab. 1, MDL-NAS model families largely outperform ordinary CNN models like ResNets [18] and RegNets [33], illustrating the visual representation potential of pure transformer models. Evidently, our MDL-NAS delivers superior results when compared to contemporary models of state-of-the-art transformers. For example, MDL-NAS-B$^\dagger$ with mask/sequential sharing policy achieves a top-1 accuracy of 82.6/82.9, surpassing DeiT-B [40] and baseline Autoformer-B [6] 0.8/1.1 and 0.2/0.5 units with comparable FLOPs and parameters. MDL-NAS-T$^\ddagger$ also yields better performance compared with its baseline Swin-T [27]. Specifically, MDL-NAS-T$^\ddagger$ with mask/sequential sharing policy outperforms Swin-T by 0.4/0.4 unit with comparable FLOPs and parameters. Through the above analysis, MDL-NAS is capable of image classification.

| Backbone | #Params | FLOPs | $\mathrm{mIoU_{ss}}$ | $\mathrm{mIoU_{ms}}$ |
|---|---|---|---|---|
| ResNet50 [18] | 67M | 951G | 42.1 | 42.8 |
| ResNet101 [18] | 86M | 1029G | 43.8 | 44.9 |
| DeiT-S [40] | 52M | 1099G | 43.2 | 43.9 |
| DeiT-B [40] | 121M | 2772G | 44.1 | 45.7 |
| S3-T [7] | 60M | 954G | 44.9 | 46.3 |
| S3-S [7] | 81M | 1071G | 48.0 | 49.3 |
| Swin-T [27] | 60M | 945G | 44.5 | 45.8 |
| Twins-S [10] | 54M | 901G | 46.2 | 47.1 |
| Focal-T [50] | 62M | 998G | 45.8 | 47.0 |
| AutoFormer-B* [6] | 112M | 1603G | 46.7 | 47.9 |
| **MDL-NAS-B† + mask** | **127M** | **1710G** | **50.1** | **50.7** |
| **MDL-NAS-B† + seq** | **130M** | **1725G** | **50.8** | **51.3** |
| **MDL-NAS-T‡ + mask** | **68M** | **993G** | **47.3** | **48.4** |
| **MDL-NAS-T‡ + seq** | **59M** | **971G** | **46.5** | **47.8** |

Table 3. ADE20K semantic segmentation. FLOPs (G) is calculated under the input scale of 512×2048.

**COCO object detection and instance segmentation.** We evaluate MDL-NAS on the COCO object detection task, where Mask RCNN [17] is applied as the basic detection framework. We report evaluation results for object detection and instance segmentation in terms of $\mathrm{AP^b}$, $\mathrm{AP^b_{50}}$, $\mathrm{AP^b_{75}}$, $\mathrm{AP^m}$, $\mathrm{AP^m_{50}}$, and $\mathrm{AP^m_{75}}$ metrics, where "b" and "m" indicate bounding box and mask metrics, respectively. $\mathrm{AP^b}$ and $\mathrm{AP^m}$ are set as the primary evaluation metrics. The comparisons between MDL-NAS and its competitors are displayed in Tab. 2. MDL-NAS achieves better performance compared with its baselines with comparable FLOPs and parameters. Specifically, regarding the bounding box metric $\mathrm{AP^b}$, MDL-NAS-B† with mask/sequential sharing policy exceeds ResNext101-64×4d [47] and the baseline Autoformer by 5.4/5.2 and 0.9/0.7 units respectively. In terms of the mask metric $\mathrm{AP^m}$, we also observe similar improvements as using mask metrics. Compared with modern methods [21,44], MDL-NAS-B† includes additional parameters due to cross-window propagation blocks, leading the parameters to be larger. In this work, instead of exploring the efficient architecture design such as Swin Transformer for sociable downstream vision tasks, we seek MDL-NAS for multiple vision tasks where they share most parameters of the backbone network, thereby promoting efficient storage deployment since multiple models can be deposited into a single one. When equipped with hierarchical vision transformer, MDL-NAS-T‡ with sequential sharing policy outperforms the typical hierarchical architecture PVT-L and Swin-T by 1.9 and 0.4 units in $\mathrm{AP^b}$, and 1.2 and 0.3 units in $\mathrm{AP^m}$. Thus, MDL-NAS is also capable of object detection.

**ADE20K Semantic Segmentation.** We also evaluate MDL-NAS on ADE20K semantic segmentation task using UperNet [46]. We report mIoU of MDA-NAS in single scale testing (ss) and multi-scale testing (ms). In Tab. 3, MDL-NAS achieves better mIoU performance than previous networks. Specifically, MDL-NAS-B† with mask/sequential sharing policy outperforms Autoformer-B by 3.4/4.1 and 2.8/3.4 units in $\mathrm{mIoU_{ss}}$ and $\mathrm{mIoU_{ms}}$ respectively. MDL-NAS-B† with mask/sequential sharing policy also achieves better performance compared with S3-S [7], demonstrating that MDA-NAS that jointly optimizes multiple vision tasks provides more benefits on semantic segmentation as the gains. Thus, MDL-NAS is also capable of semantic segmentation.

| | Top-1 Acc. | $\mathrm{AP^b}$ | $\mathrm{mIoU_{ss}}$ | #Params (t) | S-Score |
|---|---|---|---|---|---|
| Autoformer-B [6] | 82.4 | 47.3 | 46.7 | 274M | 176.4 |
| MDL-NAS-B† + mask | **82.6** | **48.2** | **50.1** | **235M** | **180.9** |
| MDL-NAS-B† + seq | **82.9** | **48.0** | **50.8** | **256M** | **181.7** |
| Swin-T [27] | 81.3 | 46.0 | 44.5 | 137M | 171.8 |
| MDL-NAS-T‡ + mask | **81.7** | **44.9** | **47.3** | **103M** | **173.9** |
| MDL-NAS-T‡ + seq | **81.7** | **46.4** | **46.5** | **115M** | **174.6** |

Table 4. The overall performance of MDL-NAS. #Params (t) denotes the total parameters for model deployment.

**Overall performance.** In this part, we compare the overall performance of MDL-NAS under these three tasks with the baselines and state-of-the-art systems, along with the total parameters for jointly optimizing these tasks. As shown in Tab. 4, MDL-NAS-B† with mask/sequential sharing policy surpasses the baseline AutoFormer 4.5/5.3 units in **S-Score** metric with much fewer parameters. MDL-NAS-T‡ with mask/sequential sharing policy also outperforms the baseline Swin-T 2.1/2.8 units in **S-Score** with 34/22M fewer parameters, further illustrating the superiority of MDL-NAS.

## 4.3. Ablation Study

In this section, we ablate important design elements in MDL-NAS for the above three vision tasks. In all ablation experiments, we finetune the supernet for these tasks with 1× training schedule in object detection for saving time.

**The effect of coarse search space.** To validate the efficacy of coarse search space, we undertake two experiments as follows: (1) after the supernet pre-training, we conduct evolutionary search to search the optimal backbone network for image classification and use the searched backbone to jointly optimize above three vision tasks, resulting in MDL-NAS-B/T-AB1 models. (2) after the supernet pre-training, we finetune the supernet to jointly optimize these vision tasks and use joint-subnet search algorithm to find optimal architectures as MDL-NAS-B/T-AB2 for all tasks. Noting that in (2), we consider two alternatives: search for the same architecture MDL-NAS-B/T-AB2(S) and different architectures MDL-NAS-B/T-AB2(D) for all tasks. The fine search space is not applied in this part and all tasks share all parameters in the backbone. From Tab. 5, MDL-NAS-B-AB2† (S/D) surpass MDL-NAS-B-AB1† 0.9/1.2 units in terms of S-Score with similar parameters, illustrating that the coarse search space can offer optimal backbone networks for all tasks rather than using the backbone designed for image classification. The same phenomenon can be observed in MDL-NAS-T. Moreover, compared with MDL-NAS-B/T-AB2† (S), MDL-NAS-B/T-AB2† (D) achieve better performance, illustrating that equipping different tasks with di-

| | Top-1 Acc. | AP$^b$ | mIoU$_{ss}$ | #Params (s) | S-Score |
|---|---|---|---|---|---|
| MDL-NAS-B-AB1$^\dagger$ | 82.9 | 43.9 | 49.6 | 55M | 176.4 |
| MDL-NAS-B-AB2$^\dagger$(S) | **82.9** | **44.6** | **49.8** | **53M** | **177.3** |
| MDL-NAS-B-AB2$^\dagger$(D) | **82.8** | **45.0** | **49.8** | **55M** | **177.6** |
| MDL-NAS-T-AB1$^\ddagger$ | 81.8 | 40.2 | 47.2 | 28M | 169.2 |
| MDL-NAS-T-AB2$^\ddagger(S)$ | **81.6** | **41.0** | **47.8** | **27M** | **170.4** |
| MDL-NAS-T-AB2$^\ddagger(D)$ | **81.6** | **41.3** | **47.8** | **27M** | **170.7** |

Table 5. The efficacy of coarse search space. #Params (s) denotes the parameters of the shared backbone.

| | Top-1 Acc. | AP$^b$ | mIoU$_{ss}$ | #Params (a) | S-Score |
|---|---|---|---|---|---|
| baseline1$^\dagger$ | 82.9 | 43.9 | 49.6 | 55M | 176.4 |
| baseline2$^\dagger$ | 83.2 | 45.6 | 48.9 | 165M | 177.7 |
| MDL-NAS-B-AB3$^\dagger$(S) + mask | **83.0** | **46.9** | **49.9** | **117M** | **179.8** |
| MDL-NAS-B-AB3$^\dagger$(D) + mask | **83.0** | **47.0** | **50.0** | **117M** | **180.0** |
| MDL-NAS-B-AB3$^\dagger$(S) + seq | **83.0** | **46.5** | **49.7** | **100M** | **179.2** |
| MDL-NAS-B-AB3$^\dagger$(D) + seq | **82.9** | **46.6** | **49.8** | **101M** | **179.3** |
| baseline1$^\ddagger$ | 81.8 | 40.2 | 47.2 | 28M | 169.2 |
| baseline2$^\ddagger$ | 82.1 | 43.5 | 43.9 | 84M | 169.5 |
| MDL-NAS-AB3-T$^\ddagger$(S)+mask | **82.0** | **43.7** | **47.2** | **45M** | **172.9** |
| MDL-NAS-AB3-T$^\ddagger$(D)+mask | **82.1** | **43.8** | **47.3** | **47M** | **173.1** |
| MDL-NAS-AB3-T$^\ddagger$(S)+seq | **82.2** | **44.3** | **46.6** | **53M** | **173.1** |
| MDL-NAS-AB3-T$^\ddagger$(D)+seq | **82.1** | **44.3** | **46.8** | **53M** | **173.2** |

Table 6. The effect of fine search space. #Params (a) means that the total parameters of all tasks in the backbone network.

verse architectures is more preferable.

**The effect of fine search space.** To validate the effectiveness of the fine search space and proposed sharing policies, we apply the same macro architecture configuration with MDL-NAS-B/T-AB1 and conduct subsequent experiments: (1) all tasks share all backbone parameters as baseline1; (2) each task monopolizes backbone parameters as baseline2; (3) we apply mask/sequential sharing policy on baseline1 and use joint-subnet search algorithm to search fine-grained share ratios for each task, where the final model is called MDL-NAS-B/T-AB3. We also consider that each task uses the same or different share ratios in each layer as MDL-NAS-B/T-AB3(S/D). As shown in Tab. 6, MDL-NAS-B/T-AB3(S/D) with mask or sequential sharing deliver better performance than baseline1 and baseline2 by a large margin with an appropriate model size, demonstrating the efficacy of fine search space and sharing policies.

## 4.4. Incremental Learning of MDL-NAS

When a new task or dataset are assigned to MDL-NAS, only the task-specific parameters are required to accommodate the new task or dataset, while all task-shared parameters are frozen. Therefore, MDL-NAS is established to allow incremental learning. In this part, we first evaluate the efficacy of MDL-NAS-B/T on CUB-200-2011 dataset [42] after MDL-NAS-B/T has been trained on classification, detection, and segmentation tasks. Next, we further assess the performance of MDL-NAS-B on the pose estimation task.

**Tune to new dataset.** After MDL-NAS-B/T has been jointly trained on the three aforementioned tasks, we employ the trained classification model as a pretrain weight to finetune the CUB-200-2011 dataset. For MDL-NAS-B/T with mask sharing policy, we consider to freeze the score parameters in Eq. (8) as MDL-NAS-B/T(S) or vary the

| | Input Size | #Params (e) | Top-1 Acc. |
|---|---|---|---|
| Swin-T [27] | 384×384 | 29.0M | 80.0 |
| Autoformer-B [6] | 384×384 | 60.6M | 80.7 |
| **MDL-NAS-B$^\dagger$(S) + mask** | **384×384** | **40.1M** | **86.8** |
| **MDL-NAS-B$^\dagger$(D) + mask** | **384×384** | **38.9M** | **86.9** |
| **MDL-NAS-B$^\dagger$ + seq** | **384×384** | **36.1M** | **86.6** |
| **MDL-NAS-T$^\ddagger$(S) + mask** | **384×384** | **7.6M** | **85.4** |
| **MDL-NAS-T$^\ddagger$(D) + mask** | **384×384** | **14.2M** | **86.6** |
| **MDL-NAS-T$^\ddagger$ + seq** | **384×384** | **11.9M** | **86.6** |

Table 7. Incremental learning experiments on CUB-200-2001 dataset. #Params (e) means that additional parameters size that each method needs when generalizing to a new task.

| | Input Size | #Params (e) | AP | AP$^{50}$ | AP$^{75}$ | AP$^M$ | AP$^L$ |
|---|---|---|---|---|---|---|---|
| Autoformer-B [6] | 256×192 | 64.4M | 70.6 | 89.3 | 79.2 | 67.7 | 76.8 |
| **MDL-NAS-B$^\dagger$(S) + mask** | **256×192** | **39.5M** | **73.5** | **92.5** | **81.6** | **70.3** | **78.0** |
| **MDL-NAS-B$^\dagger$(D) + mask** | **256×192** | **43.7M** | **73.6** | **92.5** | **81.6** | **70.7** | **78.1** |
| **MDL-NAS-B$^\dagger$ + seq** | **256×192** | **39.9M** | **74.0** | **92.5** | **82.6** | **71.5** | **78.2** |

Table 8. Incremental learning experiments on the pose estimation task.

score parameters as MDL-NAS-B/T(D) during finetuning. As shown in Tab. 7, MDL-NAS-B$^\dagger$(S/D) with mask sharing policy outperforms the baseline Autoformer by 6.1/6.2 units in Top1 Accuracy with 20.5/21.7M fewer parameters. MDL-NAS-B$^\dagger$ also surpasses Autoformer by a large margin. For MDL-NAS-T, we also observe similar improvements as MDL-NAS-B, demonstrating that MDL-NAS can support a new dataset with fewer task-specific parameters.

**Tune to new task.** For pose estimation task, we also consider MDL-NAS-B(S/D) with mask sharing policy during fitting the new task. MDL-NAS-B$^\dagger$(S/D) with mask sharing policy achieves 73.5/73.6AP, which surpasses Autoformer 2.9/3.0 units with 24.9/21.7M fewer parameters, further demonstrating the superiority of our MDL-NAS, as shown in Tab. 8. MDL-NAS-B$^\dagger$ with sequential sharing policy also outperforms AutoFormer by 3.4 units with 24.5M fewer parameters.

Based on the above analysis, MDL-NAS can be fit to a new dataset or task with a few task-specific parameters while keeping the same performances for other tasks. Training recipe and other details are given in Appendix F and J.

# 5. Conclusion

In this work, we introduce MDL-NAS, a unified framework that optimizes multiple vision tasks collectively. MDL-NAS achieves high performance for all vision tasks and keeps storage efficient for model deployment through a coarse-to-fine searching space design and a joint-subnet search algorithm. We also demonstrate that MDL-NAS can generalize to a new dataset or a new vision task with small task-specific parameters while maintaining the same performance for other vision tasks.

**Acknowledgement.** This work was supported by the National Natural Science Foundation of China (No. 62071104, No. U2233209, No. U2133211, and No. 62001063).

## References


[1] Rodrigo Berriel, Stephane Lathuillere, Moin Nabi, Tassilo Klein, Thiago Oliveira-Santos, Nicu Sebe, and Elisa Ricci. Budget-aware adapters for multi-domain learning. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 382–391, 2019. 3

[2] Deblina Bhattacharjee, Tong Zhang, Sabine Süsstrunk, and Mathieu Salzmann. Mult: An end-to-end multitask learning transformer. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 12031–12041, 2022. 1

[3] Hakan Bilen and Andrea Vedaldi. Integrated perception with recurrent multi-task neural networks. *Advances in neural information processing systems*, 29, 2016. 3

[4] Hakan Bilen and Andrea Vedaldi. Universal representations: The missing link between faces, text, planktons, and cat breeds. *arXiv preprint arXiv:1701.07275*, 2017. 3

[5] Nicolas Carion, Francisco Massa, Gabriel Synnaeve, Nicolas Usunier, Alexander Kirillov, and Sergey Zagoruyko. End-to-end object detection with transformers. In *European conference on computer vision*, pages 213–229. Springer, 2020. 1

[6] Minghao Chen, Houwen Peng, Jianlong Fu, and Haibin Ling. Autoformer: Searching transformers for visual recognition. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 12270–12280, 2021. 1, 2, 3, 4, 6, 7, 8

[7] Minghao Chen, Kan Wu, Bolin Ni, Houwen Peng, Bei Liu, Jianlong Fu, Hongyang Chao, and Haibin Ling. Searching the search space of vision transformer. *Advances in Neural Information Processing Systems*, 34:8714–8726, 2021. 3, 4, 7

[8] Zhao Chen, Vijay Badrinarayanan, Chen-Yu Lee, and Andrew Rabinovich. Gradnorm: Gradient normalization for adaptive loss balancing in deep multitask networks. In *International conference on machine learning*, pages 794–803. PMLR, 2018. 3

[9] Zhao Chen, Jiquan Ngiam, Yanping Huang, Thang Luong, Henrik Kretzschmar, Yuning Chai, and Dragomir Anguelov. Just pick a sign: Optimizing deep multitask models with gradient sign dropout. *Advances in Neural Information Processing Systems*, 33:2039–2050, 2020. 3

[10] Xiangxiang Chu, Zhi Tian, Yuqing Wang, Bo Zhang, Haibing Ren, Xiaolin Wei, Huaxia Xia, and Chunhua Shen. Twins: Revisiting the design of spatial attention in vision transformers. *Advances in Neural Information Processing Systems*, 34:9355–9366, 2021. 3, 7

[11] Xiangxiang Chu, Zhi Tian, Bo Zhang, Xinlong Wang, Xiaolin Wei, Huaxia Xia, and Chunhua Shen. Conditional positional encodings for vision transformers. *arXiv preprint arXiv:2102.10882*, 2021. 3

[12] Carl Doersch and Andrew Zisserman. Multi-task self-supervised visual learning. In *Proceedings of the IEEE international conference on computer vision*, pages 2051–2060, 2017. 3

[13] 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. *ICLR*, 2021. 3

[14] Yuan Gao, Haoping Bai, Zequn Jie, Jiayi Ma, Kui Jia, and Wei Liu. Mtl-nas: Task-agnostic neural architecture search towards general-purpose multi-task learning. In *Proceedings of the IEEE/CVF Conference on computer vision and pattern recognition*, pages 11543–11552, 2020. 3

[15] 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 *European conference on computer vision*, pages 544–560. Springer, 2020. 3

[16] Kai Han, An Xiao, Enhua Wu, Jianyuan Guo, Chunjing Xu, and Yunhe Wang. Transformer in transformer. *Advances in Neural Information Processing Systems*, 34:15908–15919, 2021. 3

[17] Kaiming He, Georgia Gkioxari, Piotr Dollár, and Ross Girshick. Mask r-cnn. In *Proceedings of the IEEE international conference on computer vision*, pages 2961–2969, 2017. 7

[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*, pages 770–778, 2016. 6, 7

[19] Ronghang Hu and Amanpreet Singh. Unit: Multimodal multitask learning with a unified transformer. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 1439–1449, 2021. 1

[20] Ronghang Hu, Amanpreet Singh, Trevor Darrell, and Marcus Rohrbach. Iterative answer prediction with pointer-augmented multimodal transformers for textvqa. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 9992–10002, 2020. 1

[21] Zilong Huang, Youcheng Ben, Guozhong Luo, Pei Cheng, Gang Yu, and Bin Fu. Shuffle transformer: Rethinking spatial shuffle for vision transformer. *arXiv preprint arXiv:2106.03650*, 2021. 6, 7

[22] Adrián Javaloy and Isabel Valera. Rotograd: Gradient homogenization in multitask learning. In *International Conference on Learning Representations*, 2022. 3

[23] Iasonas Kokkinos. Ubernet: Training a universal convolutional neural network for low-, mid-, and high-level vision using diverse datasets and limited memory. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 6129–6138, 2017. 3

[24] Liunian Harold Li, Mark Yatskar, Da Yin, Cho-Jui Hsieh, and Kai-Wei Chang. Visualbert: A simple and performant baseline for vision and language. *arXiv preprint arXiv:1908.03557*, 2019. 1

[25] Liyang Liu, Yi Li, Zhanghui Kuang, J Xue, Yimin Chen, Wenming Yang, Qingmin Liao, and Wayne Zhang. Towards impartial multi-task learning. ICLR, 2021. 3

[26] Shikun Liu, Edward Johns, and Andrew J Davison. End-to-end multi-task learning with attention. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 1871–1880, 2019. 1

[27] Ze Liu, Yutong Lin, Yue Cao, Han Hu, Yixuan Wei, Zheng Zhang, Stephen Lin, and Baining Guo. Swin transformer: Hierarchical vision transformer using shifted windows. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 10012–10022, 2021. 1, 3, 6, 7, 8

[28] Jiasen Lu, Vedanuj Goswami, Marcus Rohrbach, Devi Parikh, and Stefan Lee. 12-in-1: Multi-task vision and language representation learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 10437–10446, 2020. 1

[29] Paul Michel, Omer Levy, and Graham Neubig. Are sixteen heads really better than one? *Advances in neural information processing systems*, 32, 2019. 4

[30] Ishan Misra, Abhinav Shrivastava, Abhinav Gupta, and Martial Hebert. Cross-stitch networks for multi-task learning. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 3994–4003, 2016. 3

[31] Vladimir Nekrasov, Thanuja Dharmasiri, Andrew Spek, Tom Drummond, Chunhua Shen, and Ian Reid. Real-time joint semantic segmentation and depth estimation using asymmetric annotations. In *2019 International Conference on Robotics and Automation (ICRA)*, pages 7101–7107. IEEE, 2019. 3

[32] Zhiliang Peng, Wei Huang, Shanzhi Gu, Lingxi Xie, Yaowei Wang, Jianbin Jiao, and Qixiang Ye. Conformer: Local features coupling global representations for visual recognition. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 367–376, 2021. 1

[33] Ilija Radosavovic, Raj Prateek Kosaraju, Ross Girshick, Kaiming He, and Piotr Dollár. Designing network design spaces. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 10428–10436, 2020. 6

[34] Sylvestre-Alvise Rebuffi, Hakan Bilen, and Andrea Vedaldi. Efficient parametrization of multi-domain deep neural networks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 8119–8127, 2018. 3

[35] Ethan M Rudd, Manuel Günther, and Terrance E Boult. Moon: A mixed objective optimization network for the recognition of facial attributes. In *European Conference on Computer Vision*, pages 19–35. Springer, 2016. 3

[36] Ozan Sener and Vladlen Koltun. Multi-task learning as multi-objective optimization. *Advances in neural information processing systems*, 31, 2018. 3

[37] Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. In Yoshua Bengio and Yann LeCun, editors, *3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings*, 2015. 3

[38] Xiu Su, Shan You, Jiyang Xie, Mingkai Zheng, Fei Wang, Chen Qian, Changshui Zhang, Xiaogang Wang, and Chang Xu. Vitas: vision transformer architecture search. In *European Conference on Computer Vision*, pages 139–157. Springer, 2022. 3

[39] Ximeng Sun, Rameswar Panda, Rogerio Feris, and Kate Saenko. Adashare: Learning what to share for efficient deep multi-task learning. *Advances in Neural Information Processing Systems*, 33:8728–8740, 2020. 3

[40] Hugo Touvron, Matthieu Cord, Matthijs Douze, Francisco Massa, Alexandre Sablayrolles, and Hervé Jégou. Training data-efficient image transformers & distillation through attention. In *International Conference on Machine Learning*, pages 10347–10357. PMLR, 2021. 3, 6, 7

[41] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. *Advances in neural information processing systems*, 30, 2017. 1

[42] Catherine Wah, Steve Branson, Peter Welinder, Pietro Perona, and Serge Belongie. The caltech-ucsd birds-200-2011 dataset. 2011. 8

[43] Matthew Wallingford, Hao Li, Alessandro Achille, Avinash Ravichandran, Charless Fowlkes, Rahul Bhotika, and Stefano Soatto. Task adaptive parameter sharing for multi-task learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 7561–7570, 2022. 3

[44] Wenhai Wang, Enze Xie, Xiang Li, Deng-Ping Fan, Kaitao Song, Ding Liang, Tong Lu, Ping Luo, and Ling Shao. Pyramid vision transformer: A versatile backbone for dense prediction without convolutions. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 568–578, 2021. 3, 6, 7

[45] Teng Xi, Yifan Sun, Deli Yu, Bi Li, Nan Peng, Gang Zhang, Xinyu Zhang, Zhigang Wang, Jinwen Chen, Jian Wang, et al. Ufo: Unified feature optimization. In *European Conference on Computer Vision*, pages 472–488. Springer, 2022. 3

[46] Tete Xiao, Yingcheng Liu, Bolei Zhou, Yuning Jiang, and Jian Sun. Unified perceptual parsing for scene understanding. In *Proceedings of the European conference on computer vision (ECCV)*, pages 418–434, 2018. 7

[47] Saining Xie, Ross Girshick, Piotr Dollár, Zhuowen Tu, and Kaiming He. Aggregated residual transformations for deep neural networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1492–1500, 2017. 6, 7

[48] Weijian Xu, Yifan Xu, Tyler Chang, and Zhuowen Tu. Co-scale conv-attentional image transformers. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 9981–9990, 2021. 3

[49] Yufei Xu, Qiming Zhang, Jing Zhang, and Dacheng Tao. Vitae: Vision transformer advanced by exploring intrinsic inductive bias. *Advances in Neural Information Processing Systems*, 34:28522–28535, 2021. 1

[50] Jianwei Yang, Chunyuan Li, Pengchuan Zhang, Xiyang Dai, Bin Xiao, Lu Yuan, and Jianfeng Gao. Focal self-attention for local-global interactions in vision transformers. *arXiv preprint arXiv:2107.00641*, 2021. 7

[51] Minghao Yin, Zhuliang Yao, Yue Cao, Xiu Li, Zheng Zhang, Stephen Lin, and Han Hu. Disentangled non-local neural networks. In *European Conference on Computer Vision*, pages 191–207. Springer, 2020. 6

[52] Jiahui Yu, Pengchong Jin, Hanxiao Liu, Gabriel Bender, Pieter-Jan Kindermans, Mingxing Tan, Thomas Huang, Xiaodan Song, Ruoming Pang, and Quoc Le. Bignas: Scaling up neural architecture search with big single-stage models.

In *European Conference on Computer Vision*, pages 702–717. Springer, 2020. 3

[53] Tianhe Yu, Saurabh Kumar, Abhishek Gupta, Sergey Levine, Karol Hausman, and Chelsea Finn. Gradient surgery for multi-task learning. *Advances in Neural Information Processing Systems*, 33:5824–5836, 2020. 3

[54] Li Yuan, Yunpeng Chen, Tao Wang, Weihao Yu, Yujun Shi, Zi-Hang Jiang, Francis EH Tay, Jiashi Feng, and Shuicheng Yan. Tokens-to-token vit: Training vision transformers from scratch on imagenet. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 558–567, 2021. 3, 6

[55] Sixiao Zheng, Jiachen Lu, Hengshuang Zhao, Xiatian Zhu, Zekun Luo, Yabiao Wang, Yanwei Fu, Jianfeng Feng, Tao Xiang, Philip HS Torr, et al. Rethinking semantic segmentation from a sequence-to-sequence perspective with transformers. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 6881–6890, 2021. 1

[56] Luowei Zhou, Yingbo Zhou, Jason J Corso, Richard Socher, and Caiming Xiong. End-to-end dense video captioning with masked transformer. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 8739–8748, 2018. 1

[57] X Zhu, W Su, LW Lu, B Li, XG Wang, and Deformable DETR Dai J F. Deformable transformers for end-to-end object detection. In *Proceedings of the 9th International Conference on Learning Representations. Virtual Event, Austria: OpenReview. net*, 2021. 1