# Dynamic Inference and Top-down Attention in a Hierarchical Classification Network

André Peter Kelm[0000−0003−4146−7953], Niels Hannemann*, Bruno Heberle*, Lucas Schmidt*, Tim Rolff[0000−0001−9038−3196], Christian Wilms[0009−0003−2490−7029], Ehsan Yaghoubi[9999−0003−4146−7953], and Simone Frintrop[0000−0002−9475−3593]

Hamburg University, Hamburg, Germany
{andre.kelm@, 0hannema@informatik., bruno.heberle@, lucas.schmidt-1@studium., tim.rolff@, christian.wilms@, ehsan.yaghoubi@, simone.frintrop@}uni-hamburg.de

**Abstract.** This paper introduces a new network topology that seamlessly integrates dynamic inference cost with a top-down attention mechanism, addressing two significant gaps in traditional deep learning models. Drawing inspiration from human perception, we combine sequential processing of generic low-level features with parallelism and nesting of high-level features. This design not only reflects a finding from recent neuroscience research regarding - spatially and contextually distinct neural activations - in human cortex, but also introduces a method for generating efficient 'experts': the ability to select only high-level features of task-relevant categories. In certain cases, it is possible to bypass nearly all unnecessary high-level features, significantly reducing inference cost. We believe this paves the way for future network designs that are lightweight and adaptable, making them suitable for a wide range of applications, from compact edge devices to expansive clouds. Our proposed topology also comes with a built-in top-down attention mechanism, which allows processing to be influenced by either enhancing or inhibiting category-specific high-level features, drawing parallels to the selective attention mechanism observed in human cognition. Using targeted external signals, we experimentally enhanced predictions across all tested models/experts. In terms of dynamic inference our methodology can achieve an exclusion of up to 73.5 % of parameters and 88.7 % fewer giga-multiply-accumulate (GMAC) operations, analysis against comparative baselines show an average reduction of 40 % in parameters and 8 % in GMACs across the cases we evaluated.

**Keywords:** Adaptive Inference Efficiency · Top-down Attention Mechanism · Hierarchical Network.

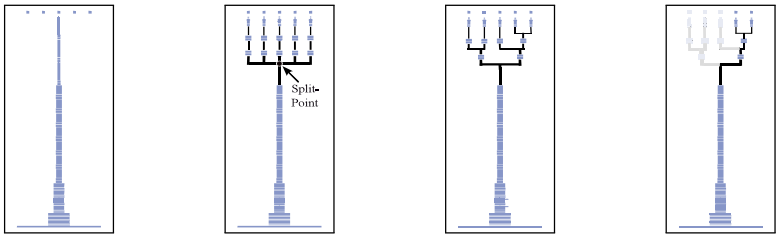
* Niels Hannemann, Bruno Heberle, and Lucas Schmidt contributed equally to this work as part of their bachelor's theses.

## 1 Introduction

One of the superior capabilities of the human brain is the ability to focus and accelerate processing when high-level knowledge is available. Not only does this save us energy, but it also increases our effectiveness and allows us to act in a very targeted manner. This perceptual ability is still challenging to achieve with current deep learning (DL) methods.

If we search for a book on the bookshelf, the salt on the table, or our friend in a crowd, the human visual system is able to focus on target-relevant features and speed up processing considerably: Wolfe's measurements of human visual perception show that the reaction speed of such a guided visual search is usually significantly higher compared to an unguided search [45]. This aspect is so general that it can also be observed across modalities. A recent neuroscience paper by Marian et al. [29], which inspired this work, argues that in the presence of audio signals such as a 'meow', which already allow semantic inferences about a searched object, humans can speed up their visual search and more quickly perceive the object, in this case, the cat.

Even with recent DL methods, such behavior is not easily reproducible [17]. The question is: How should we design artificial neural networks that have such focus and accelerated processing capability? We assume that we should be able to selectively exclude irrelevant features and include only the most relevant features in the process. However, DL methods are often described as black boxes [2], so even if a new context is available and it is clear which categories are needed for a decision and which are not, the unnecessary ones cannot be easily excluded because it is not clear at all where they are. To address this, we enhance the accessibility of high-level features by strategically restructuring them, as illustrated in Fig. 1. We transition from the conventional approach (Fig. 1a) to our proposed methodologies: the **P**arallelize features at a **H**igh-**L**evel (**PHL**) network




(a) Conventional (b) PHL (c) NHL (d) Efficient expert

Fig. 1: Neural network topologies: (blue: tensors, black: connection path). a conventional deep network from bottom to top (the dots represent five categories); (b) Our proposed parallel high-level features structure: each category has its own branch; (c) Our proposed compromise between a and b with nested high-level features; (d) Our innovative 'efficient experts' have a shallower depth than a; see depth of blue tensors (almost to scale) in the high-level region for comparison.

(Fig. 1b) and the **N**est features at a **H**igh-**L**evel (**NHL**) approach (Fig. 1c). Several neuroscientific studies validate our approach and show that the human brain also has context-specific areas [47, 39, 27]. To mimic the described biological focus our approach can be controlled extrinsically to skip task-irrelevant high-level features and directly incorporate the essentials: e.g., the super-category of marine animals to detect sharks, the super-category of plants to detect corn varieties, or cat categories to detect whether a cat is visible or not. These task-oriented configurations, called efficient 'experts' (Fig. 1d), are analogous to the 'mixture of experts' approach [36], where each expert focuses on different parts of the problem or tasks. However, our method works without gating, is category-based, and thus operates at a more granular level.

Another aspect of human perception described by Marian et al. [29] and Cheng et al. [6] assumes that there is a contextual connection between modalities and thus, for example, visual category-specific features of objects can be specifically enhanced top-down. This means that an acoustic 'meow' can be used to enhance the visual recognition of a cat [21]. DL methods such as those in Fig. 1a cannot simply integrate top-down cues to improve performance, although this would certainly be very useful [3]. However, due to their lack of interpretability [2], it's difficult to integrate top-down cues such as target or context information to mimic top-down attention [25]. This differs significantly from widely used attention mechanism, such as those used in transformer models, which tend to be bottom-up, i.e., they focus on the given data [40]. This type of attention is largely statistical and driven by numerous images/patches that the mechanism need to learn which features are more important for certain decisions than others. However, to incorporate contextual cues from the cognitive level and reinforce category-specific features, top-down attention is all you need. The topology proposed allows for categorical top-down attention, since features are categorically assigned and can now be enhanced by simple scalar weighting.

Our contributions are as follows:

- A novel expert (Fig. 1d) generation method that dynamically reduces task and computational complexity without compromising predictive performance.
- Unique hierarchical networks (Fig. 1c, variants nested with hints from a large language model) extend the method's applicability to more categories.
- A new top-down, category-based attention allows active influence on inference process of end-to-end networks or generated experts by external signals.

Overall, our method generates category-specific high-level features [48] for another step towards interpretable AI, while providing the ability to bypass unneeded high-level features to reduce parameters, computational cost, and power consumption during inference. This is important for mobile computing, industrial, drone, or robotic applications [43, 5, 42, 4], and because in industry, the cost factor of inference compared to training AI models can be up to 9:1 [12].

## 2 Related Work

### 2.1 Image Classification and Inference Efficiency

Image classification is a fundamental task in computer vision (CV). Models developed for this purpose are often adapted for various applications, including semantic segmentation, object detection, and video recognition [28]. Larger and deeper models usually offer an accuracy advantage [13], but this leads to increasing training and inference costs [12]. Notably, the industry tends to prioritize reducing inference costs over training costs by a factor of 9 [12], so a great variety of work has been done to optimize and/or dynamize inference costs [18, 41]. Cheng et al. [7] summarize strategies such as model pruning, quantization, dynamic inference costs, and efficient architectures, highlighting their efficiency, but also noting the challenge of maintaining full performance.

Our efficient expert method is intended to be orthogonal to these methods and offers the possibility of reducing the computational complexity without compromising predictive performance. Unlike other efficient methods, where efficiency is often learned within the optimization, our approach is flexible and can be controlled extrinsically (after training) to skip task-irrelevant high-level features and directly incorporate the essentials: e.g., the super-category of marine animals to detect sharks, or the super-category of plants to detect corn varieties.

### 2.2 Hierarchical Networks

Established hierarchical networks such as the HD-CNN [49] or that of Mi et al. [30] use the hierarchy between categories to make inference more efficient, but still provide only static inference. In contrast to these models, which use parallel coarse and fine classification modules or even path decision modules, our network uses a simpler and more direct approach. Since our novel topology does not require such modules, we create efficient experts by selecting only the task-relevant categories and thus dynamically adjusting the inference costs.

A recent work related to ours is the tree-like branching (TLB) network from Xue et al. [48] with a category-specific branching. They propose a tree structure based on category similarities and show improvements in image classification, but they do not perform top-down attention experiments like we do. They show improved inference costs for the whole model, but do not have the dynamic inference or efficient experts introduced here. Our architecture, its nesting, and the nesting method itself also differ significantly from [48].

### 2.3 Bottom-up vs. Top-Down Attention

Selective attention is a mechanism of human perception that enables us to focus on regions of potential interest [10, 26]. These can be objects, colors, locations, sounds, or other patterns [16]. It helps us to deal with the complexity of the world and to quickly find objects of interest. Attention is driven by two types of cues: bottom-up and top-down [38]. Bottom-up cues are salient patterns that

automatically attract attention, such as a red flower on green grass. Top-down attention, on the other hand, focuses on regions which are behaviorally relevant and is driven by pre-knowledge, goals, or expectations. The search for our key, or the experience of the well-known cocktail party effect [1], is an example of top-down attention.

Early computational attention models before the DL era have amplified target-relevant features by excitation and inhibition of pre-computed features to realize top-down attention [31, 15]. Our approach also amplifies features with top-down cues to influence the system in a goal-directed manner, but for DL. In deep networks, top-down attention is usually understood as a tracing of activations backwards through the network from category nodes to the feature maps, as in GradCAM [35], and is often used to localize category-specific features in feature maps. This requires one forward-pass of the input image through the network, before the backward tracing can be performed. However, our approach allows the inference phase of a DL method to be influenced by top-down cues to category-specific high-level features without the need for backward tracing in the first forward-pass.

## 3 Proposed Method

Section 3.1 shows our proposal to process high-level features in parallel, and that this enables category-based top-down attention. Nesting is discussed in Section 3.2, and dynamic inference by efficient experts is explained in Section 3.3.

### 3.1 High-Level Feature Parallelization

The hypothesis for splitting the architecture in parallel while processing the initial input sequentially is that the lower-level features are generic and useful for almost any category, whereas higher-level features are more specific; e.g., eye features are not required for a car and should therefore be treated in parallel. Figure 1b and 2 illustrate our **P**arallelize features at a **H**igh-**L**evel network (**PHL**), which is based on the standard ResNet50 architecture. To show the adaptability of this principle to modern architectures, we also modified a ConvNeXtV2 [46]. In these modifications, we removed the last conv block - the fifth in ResNet50 (and ConvNeXtV2) - and the FC. At this point, the architecture is split into sequential and parallel information flows. We define the transition as a 'split point', which intuitively benefits from a bottleneck structure [19] by providing more parameters to effectively manage one to $k$ connections.

*Split-Point:* Formally, we define the first part of our architecture to consist of shared network layers up to the split-point through the function: $\phi(x, \theta_0)$, taking the input image $x$ and network parameters $\theta_0$ as input. As we provide a branch for each of the target categories $k$, we define the number of final branches to be the same as our $k$ target categories. Hence, we use $k$ branch networks that we define through $\psi(\bar{x}, \theta_i)$ with $1 \leq i \leq k$, which take the output features of $\phi$ as

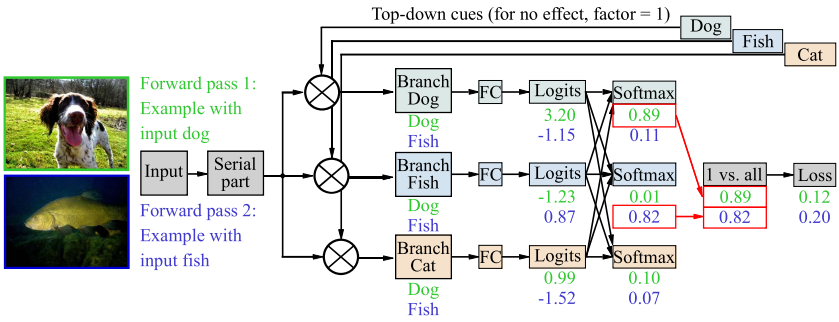



Fig. 2: PHL network with top-down attention option (inference can be influenced) and for a simple visualization with three categories: dog, fish and cat, showing some example values (green: input dog, blue: input fish). Red framed boxes indicate the selection for 1 vs. all. Our category-based top-down attention multiplies features in a given branch by a scalar during inference. For training, a neutral setting is used (multiplies each branch by one). Images from ImageNet [11].

input for $\bar{x}$. As all branch networks have the same design, we only exchange the network parameters $\theta_i$ for the classification of the $i^{\text{th}}$ class. Conv block channel depth is based on the original ResNet50 and can be found in Table 1. Using the above definitions allows us to define the output of the $i^{\text{th}}$ branch network, which is responsible for class $i$, as:

$$f_i(x) = \psi(\phi(x, \theta_0), \theta_i) = y_i. \tag{1}$$

*Cross-Entropy Loss:* For training of the network, we retain all initial target categories and estimate the class probability over all categories for the likelihood of an input $x$ belonging to a class and utilized a cross-entropy as our loss function. Note also that this is the special case of Eq. 3 where we utilized all categories instead of a subset. The result of the likelihood estimation is fed into a 1 vs. all classification approach, which selects the prediction of the correct category in a supervised manner. The cross-entropy loss is applied to this selection, penalizing incorrect classifications and rewarding correct ones for all categories; see details in Fig. 2. Optimizing the loss through backpropagation results in a high-value output for an image that belongs to the branch and a low-value output otherwise.

Table 1: Architecture - channels of conv blocks: ResNet50 vs. PHL/NHL.

| | **conv1** | **conv2** | **conv3** | **conv4** | **conv5** | | | **FC** |
|---|---|---|---|---|---|---|---|---|
| ResNet50 | 64 | 64 | 128 | 256 | 512 | | | 2048 |
| PHL/NHL | 64 | 64 | 128 | 256 | 128 | 64 | 32 | 128 |

Due to the softmax function, all paths are learned at once. Figure 2 visualizes the basic structure of our approach.

*Top-down Attention (category-specific):* The introduced PHL, with its parallel branches named by category, enables direct scalar weighting of selected category feature maps. This method is tested during validation and illustrated in Fig. 2.

### 3.2 High-Level Feature Nesting

In this section, we extend PHL to a better network topology, in which we construct nested branches depending on the categories and their superclasses. We hypothesize that there are higher-level features that should be shared, such as wheels for buses and cars, and propose the **N**est features at a **H**igh-**L**evel (**NHL**) network. A schematic of such an architecture is shown in Fig. 1c. By introducing another split-point, multiple branches $i$ share the features of a superclass $j$ through an additional sub-network $\pi(\hat{x}, \hat{\theta}_j)$ with network parameters $\hat{\theta}$ and again taking the output of $\phi$ as input for $\hat{x}$ and passing its output to $\bar{x}$ of $\psi$. Hence, a total number of superclasses $s$ results in an equal number of branches $1 \leq j \leq s$ in the lower hierarchical layer. Note that this does not change the total number of terminal branches $k$ in the upper hierarchical layer, since they are still equal to the number of categories in the dataset. This allows us to redefine the output of the hierarchical network:

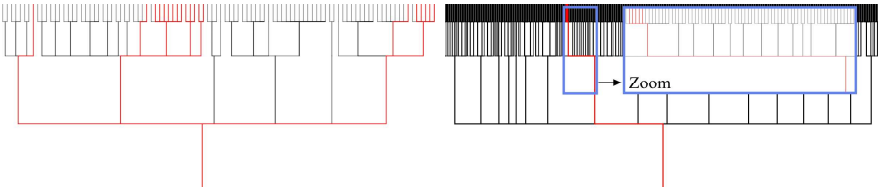
$$\hat{f}_i(x) = \psi(\pi(\phi(x, \theta_0), \hat{\theta}_j), \theta_i) = y_i. \tag{2}$$

This is repeated to get three split-points, so we extend the name to reflect the number of hierarchies: N3HL. Channels decrease with each new hierarchy level: 1) 128, 2) 64, and 3) 32 (see Table 1). This choice is intuitive, as we assume that the number of required features also decreases with each additional hierarchy level. To train the network, we use softmax and the aforementioned cross-entropy loss. To develop hierarchies in a fast and easy way, we consider several suggestions from OpenAI's ChatGPT3 to 4 [32].

*N3HL$_{\text{gpt}}$:* Since there are k categories, the number of possible nesting configurations with only two split-points can quickly become very large, exceeding $B_{100} \approx 10^{115}$ for 100 and $B_{1000} \approx 10^{1927}$ for 1000 categories, with $B_n$ being the Bell function [37]. Yet, for a network with just one split-point this number is equal to one, as it does not allow rearranging as the number of categories is fixed. Hence, for a three hierarchy levels deep network N3HL$_{\text{gpt}}$, we employed a hybrid semi-automatic approach using ChatGPT. Here, we leveraged ChatGPT whenever the classification was ambiguous.

### 3.3 Efficient Experts (Dynamic Inference)

Conventional architectures usually do not have the ability to omit category-specific high-level features that are not needed. Often it is not even clear which




(a) An expert (20 out off 100 categories). (b) An expert (5 out off 1000 categories).

Fig. 3: Select high-level features for a N3HL network with (a) 100 and (b) 1000 categories. An expert consists of the activated paths (marked in red) for the selected (a) 20 and (b) 5 categories, demonstrating the 'expert' technique (processing from bottom to top). Visualization (b) is based on the example of cross-modal interaction presented by Marian et al. in the introduction of our paper. Cognitive cues derived from a 'meow' sound lead to visual focus on cat features. The zoomed snapshot (framed in blue) shows the selection of 5 house cats from ImageNet1k and all others can be ignored, resulting in highly dynamic inference.

parameter contributes to which category. However, if only a subset of categories is needed for a specific task - whether by changed conditions or due to prior semantic knowledge - an option to exclude category-specific features would be advantageous. Through the above made definitions, we can define experts that allow the network to adapt to simpler tasks in the case that prior knowledge is available. This enables the formation of selected category-specific paths that are shorter (see Table 1) and faster than if a standard neural network has to be used (Fig. 3). An expert's categories can be selected to suit a context/task and can be selected from any level of the hierarchy as these and the categories are sorted and named. Here, we extract $n \leq k$ selected paths using a-priori knowledge of potential $n$ new target categories $C = \{i_0, \cdots, i_n \mid 1 \leq i_j \leq k\}$, while simultaneously keeping the branch fully functional. This allows us to define the probability of a sample belonging to that category $i \in C$ through:

$$p_i(x) = \frac{e^{-f_i(x)}}{\sum_{j \in C} e^{-f_j(x)}}. \tag{3}$$

Once trained, our proposal can form experts for any combination of the learned categories. ImageNet100 provides $\mathbf{2^{100} - 1}$ and ImageNet1k $\mathbf{2^{1000} - 1}$ possible experts, which do not require retraining. The ability to adapt to lower complexity of a task with less computational effort, but also to increase it when needed, is reminiscent of efficient and effective biological systems and their ability to focus.

## 4 Experiments

The NHL is generally superior to the PHL, especially for many categories, so our focus is more on the NHL with the three variants: $\text{N3HL}_{\text{gpt}}$ (see Sec. 3.2),

$N3HL152_{gpt}$ (by using ResNet152) and $N3HLCNX2_{gpt}$ (by using ConvNeXtV2tiny) [46]). We compare them with their conventional counterpart.

### 4.1 Training and Datasets

We have implemented all models in the Timm library [44] and used Rand Augment [8]. We used three different levels of classification complexity and training setups for studying PHL and NHL:

- 10 categories with 1000 epochs, batch size 32
- 100 categories with 300 epochs, batch size 56, maxed out 24,564 MebiBytes (MiB), mixed precision training
- 1000 categories with 300 epochs and batch size 24, and another run with batch size 46, maxed out 24,564 and 49,140 MiB, both trained through mixed precision training

For the datasets, subsets of [11] were selected: Imagenette and Imagewoof [20], each consisting of 10 categories. Imagenette has clearly different categories, while Imagewoof's categories are more homogeneous and consist of dog breeds. Testing these two datasets helps to understand how the parallel branches handle similar and dissimilar categories. ImageNet100 [11, 22] demonstrates the scalability of the model by a factor of 10 with a mixture of similar and dissimilar categories. The selected datasets have comparable category and image counts to Cifar10 and -100 [23, 24], which we avoided because their lower resolution introduces an unwanted new factor. ImageNet mini [14] is a subset of ImageNet1k and has only a few images per category, about 20-50, but offers a high classification complexity with 1000 categories. We focused on evaluating top-1 accuracy, excluding top-5 as it is less relevant for datasets with few categories.

### 4.2 Image Classification

Improving this task is not our main contribution, but to compare with the state-of-the-art, we conducted three experiments to evaluate our newly introduced network topology with the standard network topology currently used in established [19, 46] and leading models [33, 9], for image classification. It is necessary to compare our topology with the same models on which the modification was performed in order to specifically evaluate the impact of our topology. For this reason, there are three experiments in Table. 2, each with its own baseline:

**Experiment 1:** Using ResNet50 as a baseline, the N3HL shows better performance for 10 categories on Imagenette and -woof with a significantly lower number of parameters. For 100 categories with a similar number of parameters, the N3HL shows slightly better performance. For 1000 categories, it shows a slight drop in accuracy, probably due to overfitting due to the small amount of training data in ImageNet mini and the higher number of parameters.

**Experiment 2:** Using ResNet152 as a baseline, we are investigating how a deeper model affects our approach, and since more categories place more demands on our network topology, we have focused on the 100 and 1000 categories.

Table 2: Image classification results are divided into three table sections, each compared with its own respective baseline for fair evaluation. Experiment 1: ResNet50 compared to our N3HL$_{\text{gpt}}$. Experiment 2: Deeper models, the ResNet152 vs. N3HL152$_{\text{gpt}}$. Experiment 3: Pre-trained and modern models, the ConvNeXtV2tiny vs. our N3HLCNXV2$_{\text{gpt}}$ counterpart.

| | **Imagenette** | | **Imagewoof** | | **ImageNet100** | | **ImageNet mini** | |
|---|---|---|---|---|---|---|---|---|
| | Top-1 | param. | Top-1 | param. | Top-1 | param. | Top-1 | param. |
| *Experiment 1* | | | | | | | | |
| ResNet50 | 96.15% | 23.5 M | 90.30% | 23.5 M | 85.34% | **23.7 M** | **33.38%** | **25.6 M** |
| N3HL$_{\text{gpt}}$ | **96.18%** | **11.9 M** | **91.65%** | **11.9 M** | **85.68%** | 25.0 M | 31.78% | 113.5 M |
| *Experiment 2* | | | | | | | | **ImageNet1K** |
| ResNet152 | | | | | 86.24% | **58.3 M** | **79.05%** | **60.2 M** |
| N3HL152$_{\text{gpt}}$ | | | | | **87.08%** | 59.6 M | 78.50% | 148.2 M |
| *Experiment 3* | | | | | | | | |
| ConvNeXtV2tiny | 99.71% | 27.8 M | **96.23%** | 27.8 M | 90.53% | **27.9 M** | 81.07% | **28.6 M** |
| N3HLCNXV2$_{\text{gpt}}$ | **99.72%** | **15.3 M** | 96.22% | **15.3 M** | **90.72%** | 27.7 M | **81.10%** | 121.4 M |

For 100 categories our model has again a slightly better accuracy than the baseline. Even if the accuracy for 1000 categories is a little lower, the N3HL for ImageNet1k shows a comparatively high accuracy, which speaks for its applicability. We assume that there is still a high potential for optimization.

**Experiment 3:** This experiment complements the evaluation for two important reasons. Unlike the models trained from scratch in Experiments 1 and 2, we use frozen weights pre-trained on ImageNet21k [34], and we use a much more recent model to show whether our method is applicable here. Using ConvNeXtV2tiny [46] as a baseline, we observe for 10 categories that our method is slightly better or worse than Imagenette or -woof, but using much fewer parameters. For 100 categories, our N3HLCNXV2$_{\text{gpt}}$ has slightly better accuracy, and for 1000 categories, it also has the better accuracy. This shows that our method can be implemented on modern network architectures, and that pre-trained weights can be used effectively by replacing only the high-level part of the model with our hierarchy.

### 4.3 Efficient Experts for Dynamic Inference

The experts introduced in Sec. 3.3 are a unique feature of PHL or NHL, where parameters are reduced by focusing only on the remaining and relevant categories and can thus adapt to a changing task without retraining. For the evaluation, we do not really use specific contexts (e.g. dogs or cats), but simply show the flexibility of our method by choosing arbitrary categories. However, to show that our method can indeed focus on a specific context (here: cats), we have added an expert who does so. For the evaluation in Table 3, we have split the 10 category

Table 3: Comparison of ResNet50 and ConvNeXtV2 with our efficient experts, which can only be generated from PHL or NHL. For Imagenette and -woof we have two experts (upper and lower categories) per dataset (separated by dashed line for visualization purposes) and for ImageNet100 and ImageNet1k one each.

| | **Imagenette** | | **Imagewoof** | |
|---|---|---|---|---|
| **Metric** | **ResNet50** | **PHL** | **ResNet50** | **N3HL$_{\text{gpt}}$** |
| GMACs | 4.13 | **3.99** | 4.13 | **3.91** |
| GMACs of expert | N/A | **3.66** | N/A | **3.73** |
| GMACs reduction | 0% | **-8.40%** | 0% | **-4.90%** |
| Parameter | 23.5 / 23.5M | **20.0 M** | 23.5 / 23.5 M | **11.9 M** |
| Parameter of expert | N/A | **14.3 M** | N/A | **10.8 M** |
| Parameter reduction | 0% | **-28.63%** | 0% | **-9.63%** |
| Train categories | upper 5 / 10 | 10 | upper 5 / 10 | 10 |
| Val categories | upper 5 | upper 5 | upper 5 | upper 5 |
| Top-1 acc. | 97.16 / **97.83%** | 97.62% | 91.18 / 91.88% | **92.69%** |
| Train categories | lower 5 / 10 | 10 | lower 5 / 10 | 10 |
| Val categories | lower 5 | lower 5 | lower 5 | lower 5 |
| Top-1 acc. | 96.79 / **98.04%** | **98.04%** | **96.23**/ 94.30% | 95.89% |

| | **ImageNet100** | | **ImageNet1k** | |
|---|---|---|---|---|
| **Metric** | **ResNet50** | **N3HL$_{\text{gpt}}$** | **ConvNeXtV2** | **N3HLCNX2$_{\text{gpt}}$** |
| GMACs | **4.13** | 5.86 | **4.46** | 14.55 |
| GMACs of expert | N/A | 4.22 | N/A | **3.87** |
| GMACs reduction | 0% | **-28%** | 0% | **-73.4%** |
| Parameter | **23.5 / 23.7M** | 25M | **27.8 / 28.6M** | 119.9M |
| Parameter of expert | N/A | **13.8M** | N/A | **13.5M** |
| Parameter reduction | 0% | **-44.5%** | 0% | **-88.7%** |
| Train categories | 20 / 100 | 100 | 5 / 1000 | 1000 |
| Val categories | 20 | 20 | 5 | 5 |
| Top-1 acc. | 88.5 / 90.6% | **93.4%** | **80.0** / 79.6% | **80.0%** |

datasets 50:50 into two subsets: lower label categories (e.g., category-ID: 1-5) and upper label categories (e.g., category-ID: 6-10) to get two experts for one dataset. For the 100 categories, we split 20:80, and for the 1000 categories, we split 5:1000 and evaluated one expert each. For creating some perfectly fitting baselines, we have ResNets that are trained like the experts on the full dataset and validated only on the subset, and ResNets that are trained from scratch only on the subset.

For Imagenette upper and lower 5 both experts have a better top-1 acc. than newly trained ResNets. Slightly better are the ResNets, also trained on 10 categories, but the experts are able to reduce the parameters by approximately 28 %. To determine the computational complexity, we count giga-multiply-accumulate operations (GMACs) and find that we can save about 8 % while maintaining very high performance. In Imagewoof, the two experts have a top-1 acc. with

high values for the upper and a little worse for the lower 5, but N3HL$_{\mathbf{gpt}}$ already requires fewer parameters and GMACs for 10 categories, and even with the additional parameter and GMAC reduction for the experts, it gives competitive performance.

For ImageNet100, we show an expert of 20 random categories, the same expert used in Fig. 3a. It has only a small increase in GMACs and gives a reduction of about 10 million parameters, but outperforms the baselines by some percentage of top-1 accuracy.

The last expert in the lower right corner of Table 3 shows an evaluation for ImageNet1k. It is a good result for our method in terms of predictive performance, since it achieves the same or slightly better accuracy than baselines, with fewer GMACs, fewer parameters, and less depth.

But most importantly, this expert builds on the example of semantic inference by Marian et al. [29] introduced in Sec. 1. Our approach can use previously interpreted semantic cues, such as the sound 'meow', to focus visually. This allows the model to efficiently narrow down ImageNet's 1000 learned categories to the 5 relevant ones representing house cats (Fig. 3b). This demonstrates the practical application of our method with a high dynamic range of inference costs. These task-specific adjustments demonstrate the adaptability of our method over baselines that are not applicable (N/A) in this a way.

### 4.4 Top-Down Attention

Consider a task where images are to be classified and other modalities such as audio or descriptions are available, already interpreted, and provide prior semantic knowledge. These contextual cues can positively influence here, e.g. a 'meow' strengthens the recognition of the object cat as in human perception [45].

We argue in Sec. 3.1 that the proposed PHL/NHL and their experts has a built-in selective attention mechanism because we know where to access the category-specific high-level features, since feature maps can be associated with a branch, its filters/parameters, and a category. Figure 4 visualizes the feature maps of two parallel branches, showing category-specific differences. In an initial test, we assume that meaningful semantic knowledge already exists for each validation image, so that amplification should be done for each image category. This is achieved by multiplying the feature maps by a scalar in the corresponding path, just before the final conv layer in conv block 5 of PHL/NHL. The values are determined by experimentation and listed in Table 4, as there is a value at which the result can be influenced to the maximum; as soon as this value is exceeded, the influence ebbs away again. N3HLCNX2$_{\text{gpt}}$ behaves differently; here, higher values still lead to improvements, but only slight ones above a certain level, so we chose a value of 100, as no significant changes occur above this level. For evaluation, the labels are used to point to the category-specific path of the respective validation image in order to multiply the category-specific feature map layer with the scalar. This is done for all validation images and the first section in Table 4 shows that all classification results of our models/experts are positively influenced. In Table 5, we simulate a more realistic scenario where

Table 4: Impact of top-down cues on top-1 acc. across different datasets for our PHL, $N3HLCNX2_{gpt}$ and two efficient experts.

| | **Imagenette** | **Imagewoof** | **ImageNet100** | **ImageNet mini** |
|---|---|---|---|---|
| Top-1 PHL | 95.71% | 89.97% | 80.52% | 21.10% |
| Top-down att. signal | 1.8 | 1.4 | 1.1 | 1.3 |
| Top-1 improvement | +0.67% | +0.57% | +0.04% | +0.24% |
| | **ImageNet100** | | **ImageNet1k** | |
| Model | $N3HL_{gpt}$ | $N3HLCNX2_{gpt}$ | $N3HLCNX2_{gpt}$ | $N3HLCNX2_{gpt}$ |
| Expert | yes (20 of 100) | no (full model) | no (full model) | yes (5 of 1000) |
| Top-down att. signal | 1.8 | 100 | 100 | 100 |
| Top-1 improvement | +0.80% | +0.48% | +3.47% | +1.60% |

Table 5: Impact of incorrect top-down cues (all have the same factor of 100) on top-1 acc. of $N3HLCNX2_{gpt}$ (full model) on ImageNet1k.

| Wrong branches | 0 % | 10 % | 25 % | 50 % | 75 % | 90 % | 99.9 % | 100 % |
|---|---|---|---|---|---|---|---|---|
| Top-1 acc. | 84.57 % | 84.25 % | 83.73 % | 82.90 % | 81.98 % | 81.60 % | 81.10 % | 80.08 % |
| Improvement | +3.47 % | +3.15 % | +2.63 % | +1.80 % | +0.88 % | +0.50 % | +0.0 % | -0.02 % |

top-down cues occasionally reinforce incorrect branches. We successively increase these incorrect assignments and see a decrease in the original improvement, but our model remains robust so that even with up to 99.9 % incorrect assignments, the top-1 acc. is never falling below the original.

This novel built-in top-down attention should be further explored, as it's not available in conventional architectures without much more complex operations [25]; we therefore suggest considering hierarchical networks for cognitive computing and models with category-specific top-down attention.

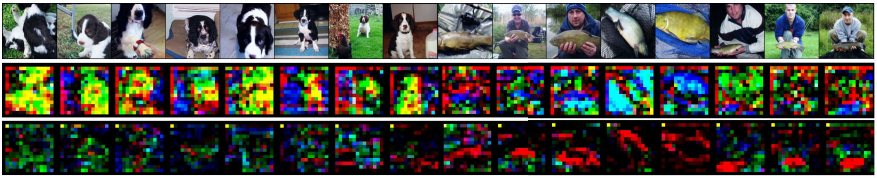


Fig. 4: Category-specific features and original images from ImageNet [11]. The top row displays the images at a reduced resolution. The second and third rows show $14 \times 14 \times 3$ RGB-colored, category-specific high-level features extracted using PHL - first parallel conv layer. Specifically, the features in the second row are from the 'dog' branch, while the ones in the third row from the 'fish' branch. The selected feature maps from the dog branch are salient for the dog object and less salient for the fish object. The opposite is observed for the fish branch.

## 5 Conclusion

In this study, we have introduced a novel network topology that seamlessly integrates dynamic inference cost with a top-down attention mechanism. Inspired by the perceptual capabilities of the human brain, we combine traditional sequential processing of low- and mid-level features with innovative parallel processing and nesting of high-level features. Since this is a basic principle, it should be possible to apply it to the latest, already effective and/or unsupervised methods.

In terms of dynamic inference cost our method can achieve an exclusion of up to 73.48 % of parameters and 88.7 % fewer GMACs, analyses with comparable baselines show an average reduction of 40 % in parameters and 8 % in GMACs across the cases we evaluated, without compromising predictive performance.

Our experiments indicate that our method paves the way for AI models that are more interpretable, energy efficient, adaptive, and do not compromise on predictive performance.

This advancement holds substantial promise for mobile computing, industrial, drone, robotic, and edge device applications, where computational resources are often limited. Future research will aim to further refine these topologies, potentially leading to breakthroughs in AI that more closely resemble human cognitive processes.

**Acknowledgements** Funded by the Deutsche Forschungsgemeinschaft (DFG, German Research Foundation) in the project Crossmodal Learning, TRR 169.

## References

1. Arons, B.: A review of the cocktail party effect. Journal of the American Voice I/O Society **12**(7) (1992)
2. Aytekin, C.: Neural networks are decision trees. ArXiv, arXiv:2210.05189 (2022)
3. Banik, S., Lauri, M., Knoll, A., Frintrop, S.: Object localization with attribute preference based on top-down attention. In: Computer Vision Systems. Springer International Publishing (2021)
4. Budiharto, W., Gunawan, A.A.S., Suroso, J.S., Chowanda, A., Patrik, A., Utama, G.: Fast object detection for quadcopter drone using deep learning. In: 2018 3rd International Conference on Computer and Communication Systems (ICCCS) (2018)
5. Cai, H., Lin, J., Lin, Y., Liu, Z., Tang, H., Wang, H., Zhu, L., Han, S.: Enable deep learning on mobile devices: Methods, systems, and applications. ACM Trans. Des. Autom. Electron. Syst. **27**(3) (2022)
6. Chen, Y.C., Spence, C.: When hearing the bark helps to identify the dog: semantically-congruent sounds modulate the identification of masked pictures. Cognition **114**(3) (2009)
7. Cheng, H., Zhang, M., Shi, J.Q.: A survey on deep neural network pruning-taxonomy, comparison, analysis, and recommendations (2023)
8. Cubuk, E.D., Zoph, B., Shlens, J., Le, Q.V.: Randaugment: Practical automated data augmentation with a reduced search space. In: IEEE/CVF CVPRW (2020)

9. Dai, Z., Liu, H., Le, Q.V., Tan, M.: Coatnet: Marrying convolution and attention for all data sizes. In: Ranzato, M., Beygelzimer, A., Dauphin, Y., Liang, P., Vaughan, J.W. (eds.) NeurIPS. vol. 34, pp. 3965–3977. Curran Associates, Inc. (2021)
10. Davis, E.T., Palmer, J.: Visual search and attention: an overview. Spatial Vision (2004)
11. Deng, J., Dong, W., Socher, R., Li, L.J., Li, K., Fei-Fei, L.: Imagenet: A large-scale hierarchical image database. In: IEEE CVF CVPR (2009)
12. Desislavov, R., Martínez-Plumed, F., Hernández-Orallo, J.: Trends in ai inference energy consumption: Beyond the performance-vs-parameter laws of deep learning. Sustainable Computing: Informatics and Systems **38** (2023)
13. Fang, Y., Wang, W., Xie, B., Sun, Q., Wu, L., Wang, X., Huang, T., Wang, X., Cao, Y.: Eva: Exploring the limits of masked visual representation learning at scale. In: 2023 IEEE/CVF CVPR. IEEE Computer Society (2023)
14. Figotin, I.: ImageNet 1000 (mini) (2020), available at Kaggle: https://www.kaggle.com/datasets/ifigotin/imagenetmini-1000
15. Frintrop, S., Backer, G., Rome, E.: Goal-directed search with a top-down modulated computational attention system. In: Pattern Recognition: 27th DAGM Symposium. Proceedings 27. Springer (2005)
16. Frintrop, S., Rome, E., Christensen, H.I.: Computational visual attention systems and their cognitive foundations: A survey. ACM Transactions on Applied Perception (TAP) **7**(1) (2010)
17. Fu, D., Weber, C., Yang, G., Kerzel, M., Nan, W., Barros, P., Wu, H., Liu, X., Wermter, S.: What can computational models learn from human selective attention? a review from an audiovisual unimodal and crossmodal perspective. Frontiers in Integrative Neuroscience **14** (2020)
18. Han, Y., Huang, G., Song, S., Yang, L., Wang, H., Wang, Y.: Dynamic neural networks: A survey. IEEE TPAMI **44**(11) (2022)
19. He, K., Zhang, X., Ren, S., Sun, J.: Deep residual learning for image recognition. In: 2016 IEEE CVPR (2016)
20. Howard, J.: Imagewang. https://github.com/fastai/imagenette/
21. Iordanescu, L., Grabowecky, M., Franconeri, S., Theeuwes, J., Suzuki, S.: Characteristic sounds make you look at target objects more quickly. Attention, Perception, & Psychophysics **72**(7) (2010)
22. kaggle: Imagenet100. https://www.kaggle.com/datasets/ambityga/imagenet100, accessed: 2023-11-17
23. Krizhevsky, A., Nair, V., Hinton, G.: Cifar-10 (canadian institute for advanced research)
24. Krizhevsky, A., Nair, V., Hinton, G.: Cifar-100 (canadian institute for advanced research)
25. Kuo, T.Y., Liao, Y., Li, K., Hong, B., Hu, X.: Inferring mechanisms of auditory attentional modulation with deep neural networks. Neural Comput. **34**(11) (2022)
26. Lev-Ari, T., Beeri, H., Gutfreund, Y.: The ecological view of selective attention. Frontiers in Integrative Neuroscience **16** (2022)
27. Liu, K.Y., Li, X.Y., Lai, Y.R., Su, H., Wang, J.C., Guo, C.X., Xie, H., Guan, J.S., Zhou, Y.: Denoised internal models: A brain-inspired autoencoder against adversarial attacks. Machine Intelligence Research **19**(5) (2022)
28. Liu, Z., Ning, J., Cao, Y., Wei, Y., Zhang, Z., Lin, S., Hu, H.: Video swin transformer. In: 2022 IEEE/CVF CVPR. IEEE Computer Society (2022)
29. Marian, V., Hayakawa, S., Schroeder, S.R.: Cross-modal interaction between auditory and visual input impacts memory retrieval. Frontiers in Neuroscience **15** (2021)

30. Mi, J.X., Li, N., Huang, K.Y., Li, W., Zhou, L.: Hierarchical neural network with efficient selection inference. Neural Netw **161** (2023)
31. Navalpakkam, V., Itti, L.: Modeling the influence of task on attention. Vision research **45**(2) (2005)
32. OpenAI: Chatgpt. https://openai.com/ (2023/24)
33. Pham, H., Dai, Z., Xie, Q., Le, Q.V.: Meta pseudo labels. In: CVPR. pp. 11552–11563 (2021)
34. Ridnik, T., Baruch, E.B., Noy, A., Zelnik, L.: Imagenet-21k pretraining for the masses. In: Vanschoren, J., Yeung, S. (eds.) Proceedings of the NeurIPS Datasets and Benchmarks (2021)
35. Selvaraju, R.R., Cogswell, M., Das, A., Vedantam, R., Parikh, D., Batra, D.: Grad-cam: Visual explanations from deep networks via gradient-based localization. In: Proceedings of the IEEE ICCV (2017)
36. Shazeer, N., Mirhoseini, A., Maziarz, K., Davis, A., Le, Q.V., Hinton, G.E., Dean, J.: Outrageously large neural networks: The sparsely-gated mixture-of-experts layer. In: ICLR. OpenReview.net (2017)
37. Tanny, S.M.: On some numbers related to the bell numbers. Canadian Mathematical Bulletin **17**(5) (1975)
38. Theeuwes, J.: Top–down and bottom–up control of visual selection. Acta psychologica **135**(2) (2010)
39. Tonegawa, S., Liu, X., Ramirez, S., Redondo, R.: Memory engram cells have come of age. Neuron **87**(5) (2015)
40. Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A.N., Kaiser, L.u., Polosukhin, I.: Attention is all you need. In: NeurIPS. vol. 30. Curran Associates, Inc. (2017)
41. Wang, H., Zhang, W., Su, S., Wang, H., Miao, Z., Zhan, X., Li, X.: Sp-net: Slowly progressing dynamic inference networks. In: ECCV. Springer Nature Switzerland (2022)
42. Wang, J., Cao, B., Yu, P., Sun, L., Bao, W., Zhu, X.: Deep learning towards mobile applications. In: 2018 IEEE 38th International Conference on Distributed Computing Systems (ICDCS) (2018)
43. Wang, Y., Wang, J., Zhang, W., Zhan, Y., Guo, S., Zheng, Q., Wang, X.: A survey on deploying mobile deep learning applications: A systemic and technical perspective. Digital Communications and Networks **8**(1) (2022)
44. Wightman, R.: Pytorch image models. https://github.com/rwightman/pytorch-image-models (2019)
45. Wolfe, J.M.: Guided search 6.0: An updated model of visual search. Psychonomic Bulletin & Review **28**(4) (2021)
46. Woo, S., Debnath, S., Hu, R., Chen, X., Liu, Z., Kweon, I.S., Xie, S.: Convnext v2: Co-designing and scaling convnets with masked autoencoders. In: IEEE/CVF CVPR (2023)
47. Xie, H., Liu, Y., Zhu, Y., Ding, X., Yang, Y., Guan, J.S.: In vivo imaging of immediate early gene expression reveals layer-specific memory traces in the mammalian brain. Proceedings of the National Academy of Sciences **111**(7) (2014)
48. Xue, M., Song, J., Sun, L., Song, M.: Tree-like branching network for multi-class classification. In: Intelligent Computing & Optimization. Springer International Publishing (2022)
49. Yan, Z., Zhang, H., Piramuthu, R., Jagadeesh, V., DeCoste, D., Di, W., Yu, Y.: Hd-cnn: Hierarchical deep convolutional neural networks for large scale visual recognition. In: IEEE ICCV (2015)