Modularization as a Leverage to Facilitate Trustworthiness of Neural Networks

someone 1
a university
somewhere in the world
Email: someone1@xyz.com

someone 2
a university
somewhere in the world
Email: someone2@xyz.com

***Abstract*—The progress in Deep Learning (DL) in recent years has led to many new, previously inaccessible fields of activity now being opened up. At the same time, however, the requirements in aspects such as scalability, transparency and trust have increased as well. To address this, we propose modularization as a technique to reduce hardware requirements for training and inference of models and to facilitate knowledge transfer, error traceability and interpretability / transparency of predictions. In this work, we focus on the latter two aspects and examine the central question whether the approach can be utilized to incentivize neural networks to learn conceptual relationships between the different categories and hence increase the trustworthiness of predictions. We restrict our considerations to the most common forms of modularization and assess the approach on a qualitative basis. Using an augmented version of the CIFAR100 dataset, we experimentally investigate the impact of modularization on the predictive conceptual proximity of CNNs. The results indicate that by means of modularization a good compromise between trustworthiness and performance can be achieved.**

I. Introduction

The progress in image recognition in the last two decades has been nothing short of astonishing, not least due to the success of CNNs ([1]). However, as a consequence of the ongoing endeavor to solve evermore complex tasks, many modern models can't be economically feasibly trained anymore without computing capacities that necessitate access to datacenters or cloud platforms. Being neither a completely novel nor in itself negative development, it does effect a drastic reduction in practical applicability and facilitates an increasing oligopolization of the Deep Learning industry as well as research in favor of those with access to more resources and / or data. This is further catalyzed by the inherent limitation of DL regarding the impossibility to refactor once learnt knowledge and the requirement to retrain the whole model each time a shift in the data base occurs.

Moreover, the decision-making process of conventional neural networks is difficult to comprehend due to their inherent nature. The latter is, however, becoming more and more important as such systems are increasingly employed in safety-critical applications such as autonomous driving or for predictions with high economic implications. It is undisputed that Trustworthy AI is a prerequisite for the public acceptance of DL.

To address these shortcomings, we propose the modularization of neural networks in abstraction of the well-known "divide and conquer" principle from computer science. This work is part of a wider study on this topic, focusing on the aspect of interpretability and model trustworthiness in particular.

Specifically, we argue that by explicitly inferring the structure of the label space into the model architecture, the network is incentivized to learn conceptual relationships between the different categories. In other words, predictions by a modularized network are on average conceptually closer to the ground truth than those of conventional neural networks.

As a positive ancillary effect, decomposition simplifies the interpretability of the model's decision making process since humans equally tend to decompose complex problems into simpler sub-problems; hence, forecasts based on several iterative sub-decisions are easier to comprehend than monolithic predictions without intermediate results.

While the presented concepts are theoretically transferable to arbitrary models, in this work we limit our focus to Directed Acyclic Graphs, in particular CNNs as a subcategory of neural networks.

In summary, this paper has the following contributions: We first discuss advantages and disadvantages of the presented approach. Subsequently, a general theoretical foundation regarding the modularization of neural networks is given. Specifically, different ways to decompose a network are analyzed, the impact of modularization on convergence and performance properties of a model are discussed and the concept of modularization error is introduced.

In a dedicated section we debate different aspects of trustworthy AI and how to measure them. In this regard, we extend the concept of semantic distance established by [2] to be able to effectively compare different models on this basis.

We do then present two experiments in which we compare a modularized network and a comparable conventional CNN on an augmented version of the CIFAR100 dataset to assess how the presented technique affects the model's behavior with regard to its predictive conceptual proximity. Our findings suggest that, at the expense of a slight decrease of absolute performance, modularization does indeed incentivize the model to learn conceptual relationships, resulting in more trustworthy predictions. The work is concluded by a summarizing discussion of the results.

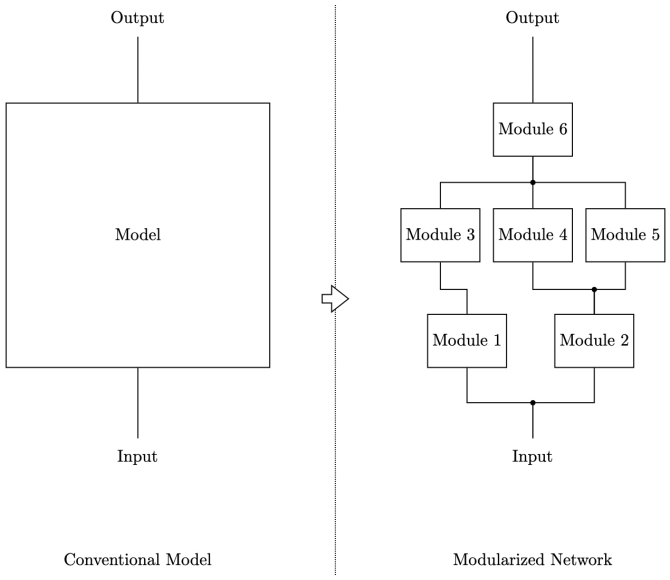

Fig. 1. The basic idea: By explicitly inferring structural information into the model, modularization facilitates behavioral comprehensibility of neural networks and incentivizes conceptual proximity of predictions.

II. Related Work

The idea of composing a single model from multiple submodules is not a novel one; Initial approaches to combine multiple learners dating back to the late 80s originated from the domain of competitive learning, mainly born as an approach to overcome the limitations of computational resources that served as the primary bottleneck in machine learning (ML) at that time.

Early works such as [3] and [4] develop "divide and conquer" strategies to have different networks compete for subspaces of the domain, facilitated by gating networks that learn to dynamically switch between the submodules depending on the input. Adopting a similar philosophy, [5] derive a stochastic framework for the "divide and conquer" approach; i. e. the model is trained using a variant of the EM (Expectation Maximization) algorithm to learn a mixture of probabilities from which the final result is then drawn. The authors argue that such an approach not only decreases hardware requirements but also increases learning speed and interpretability, reduces spatial and temporal crosstalk and leads to better generalization.

Early general studies on consensus theory in non-competitive modular systems can be found in [6] and [7]. These works, however, focus on the combination of different independently trained learners to reduce uncertainty and error rates when compared to the individual composing submodules. An overview of the fields in which modularization is employed for task-independent performance boosting today can e. g. be found in [8].

Concerning task-specific developments leveraging modularization, especially in image recognition, composed approaches have become increasingly popular in research since the establishment of publicly accessible, large-scale datasets with multiple label hierarchy levels such as CIFAR100 or ImageNet in the last decade.

E. g. [9] propose a network composed of multiple template models. They employ unsupervised learning techniques to identify natural clusters of alike classes and train one building block model for each meta-concept found this way. Thereby the authors aim to maximize the decision boundaries between categories and optimize the usage of modeling capacity per individual module. Their work suggests the validity of the approach and devises different techniques to further reduce hardware requirements for composed models.

A very recent publication originating from a different point of view by [10] employs tree-like, dynamically growing structures of CNNs to combat what the authors denote as "catastrophic forgetting" in image recognition, i. e. networks forgetting learned patterns when they are retrained on new data. Furthermore, the authors argue that a modular structure facilitates expendability and adaptability of DL models for situations where new data becomes steadily available or a constant distributional shift takes place as is the case in many real-world scenarios.

Most recent findings in the form of [11] furthermore suggest that neural networks in general tend to naturally form modular clusters when being trained using modern regularization techniques such as Pruning or Dropout and might thus be inherently accessible to modularization.

III. Advantages of a Modular Model Structure

In theory, modular design introduces several desirable characteristics with regard to practical applicability of neural networks. In the following, a brief overview of the different benefits of modularization is given.

Firstly, a network composed from multiple submodules has the advantage of providing several breakpoints throughout the decision making process. It de facto breaks up the black box that is a monolithic model and provides insights in how the final prediction is formed, thus facilitating comprehensibility of the model's behavior. We hypothesize that part of the reason for the above is that this kind of incremental reasoning is closer to the human decision making process, similar to how we also naturally tend to break down complex problems in smaller, simpler ones. Hence, forecasts based on several iterative sub-decisions are easier to comprehend than monolithic predictions without intermediate results.

Secondly, being able to trace the decision making process throughout the network through the interim results of the different submodules equally facilitates traceability of errors and identification of error sources in general. I. e. by tracing the flow of information through the composed network, it is possible to perform post-mortem analysis of error causality and location. While the former provides an understanding of the nature of the error, the latter indicates where exactly adjustments have to be made to fix this deficit. Both are highly desirable characteristics in practice, especially for applications with safety implications that necessitate deterministic traceability of failures. Moreover, a modular structure allows for the

incorporation of deterministic modules such as error detection mechanisms or fallback routines in addition to conventional network elements (in the sense of "hybrid models") which can contribute to achieving predictable model behavior and ensuring trustworthiness.

Other benefits of modularization include enhanced retrainability and modifiability / extensibility of models, simplified refactoring of once-learnt knowledge – either through transfer learning or the outright reuse of entire submodules in different composed networks – and facilitated accessibility of ML in hardware constrained environments by reduction of the minimal necessary computing power for training and inference.

Concluding, modularization offers various benefits in comparison to conventional monolithic networks, in particular flexibility / modifiability and trustworthiness / interpretability, the latter being the focus of this work.

IV. Composition of Meta-Networks from Multiple Submodules

A. Decomposable Dimensions of Neural Networks

To resolve the ambiguity of the term modularization in the context of ML, we begin by identifying the different dimensions of neural networks to which the technique is applicable.

Consider an arbitrary ML model. While seeming like a single, cohesive system on the surface, it is in reality constituted from multiple different dimensions characterizing the concrete instantiation of the model:

- **Task:** A task is an algorithmic description of a particular process relating a specific input to a specific output that can be expressed by means of mathematical or natural language statements. Every particular model instantiation is coupled to a specific, existence-giving task. It is the purpose of the model to learn the transformation defined by the latter.
- **Model:** Model in this case refers to the functional depiction of the pivotal task as well as all related subtasks. This includes aspects such as the concrete type and architecture of the employed model, preprocessing routines, training and testing procedures, etc.
- **Implementation:** Ultimately, every model is a piece of software. It can be interpreted as a projection of the functional model to the computational level. While not inferring any information on its own, the implementation very well affects the overall system in aspects such as (computational) performance, modifiability, etc.

Modularization as a technique is applicable to any of the aforementioned dimensions and yields different effects depending on the context (cf. figure 2).

For instance, the modeled task can be separated into several sub-tasks based on its inherent nature, that is, if it can be interpreted as a meta-task composed of multiple smaller tasks. The model layer directly reflects such decompositions on the task layer. However, within these constraints, modularization can occur independent of the considered process as well, e. g. motivated by the inherent structure of the underlying data base.

Both modularization on the task as well as on the model layer infer information augmenting the topology of the search space into the model. We denote this as auxiliary information in contrast to primary information, i. e. knowledge contained in the relation between domain and codomain. Specifically, modularization enhances the structure of the search space by introducing additional constraints, effectively dividing it into several sparsely connected subspaces, each of which is of lower complexity than the composed space. The amount of auxiliary information inferred by a particular decomposition is proportional to the amount as well as the strictness (i. e. holonomic or non-holonomic) of the introduced constraints. Consequently, the existing modeling capacity can be utilized more efficiently as the amount of information the model has to learn explicitly is reduced.

The increased specificity of the model, however, results in a reduction of the generalizability at the same time as described by the "No Free Lunch" theorem by [12]. Hence modularization is always case specific and can hardly be transferred between tasks. I. e. while the submodules can easily be reused in other applications, the individual decomposition has to be conceived anew for each new model.

The software layer on the other hand does not infer any information into the system. It can thus be designed in an arbitrarily modularized way as long as the projection of the functional model is retained. As mentioned before, this may still impact aspects such as modifiability or computational performance.

B. Types of Modularization

There are two fundamental ways to decompose a system into submodules, horizontal and vertical decomposition. Note that with "system", we denote any structure or process that transforms an arbitrary input into an equally arbitrary output here. Both types of modularization act on the transformation from domain to codomain and are not mutually exclusive, i. e. they are freely combinable.

Horizontal Modularization: Horizontal modularization describes the process of decomposing an arbitrary process in a suitable way to approximate a parental process through multiple parallel (optimally independent) sub-processes. From a mathematical point of view, the decomposition can be understood as an optimization problem where a transformation from a certain domain to a codomain is to be separated in a way that the informational loss in comparison to the original transformation is to be minimized. For an optimal decomposition, the resulting modules are capable of modelling the primary information in its entirety; the auxiliary information inherently increases due to the additional holonomic constraints inferred by the separation into individual submodules. In consequence, the resulting meta-network learns faster but not necessarily better. Three forms of horizontal modularization are possible:

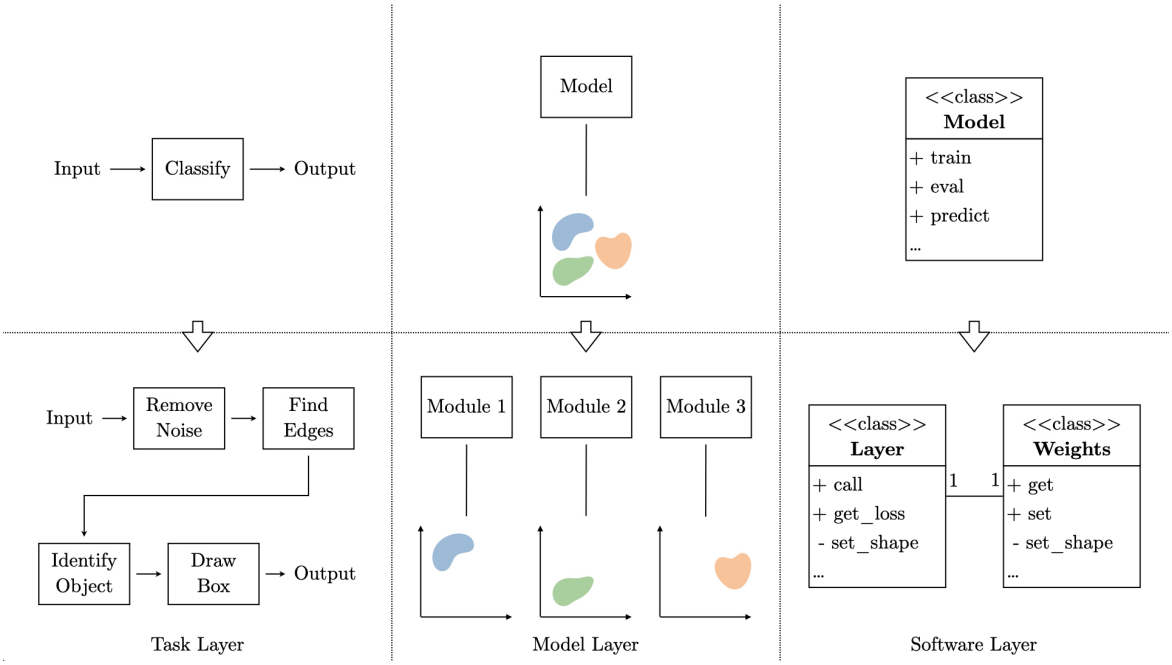

Fig. 2. Modularization can be applied to any of the three domains constituting a specific model instantiation.

- Decomposition in the domain with unchanged codomain and reconstruction of the original transformation via aggregation of the results of the different submodules
- Decomposition in the codomain with unchanged domain and reconstruction of the original transformation via merging / superimposing of the subspaces of the different submodules
- Arbitrary combinations of the former two

Vertical Modularization: Vertical modularization, also known as Layering, describes the process of modulating the information channel along which the primary information in the domain is mapped to the codomain by decomposing a process into different sub-processes, also called Layers, along the axis of transformation. The amount of primary information in the system is restricted by the maximum amount the informationally most limited layer can store. In other words, if e. g. any of the intermediate layers in a modular system possesses a single Boolean as sole output, it is impossible for outputs of subsequent layers (and thus the overall model) to be of higher complexity without additional input. Note, however, that the amount of primary information a layer can contain is composed of the cardinality as well as the interrelations of the states within the layer. Thus a pure dimensionality reduction from one layer to the next does not necessarily imply loss of information. Similar to horizontal modularization, vertical modularization increases the auxiliary information and hence the specificity of the system.

Popular applications leveraging vertical modularization e. g. include contemporary Natural Language Processing architectures that chain neural networks and autoencoders.

C. Convergence and Performance Properties

An interesting and certainly legitimate question in regard to composed networks is whether, just because convergence was guaranteed for a given model before modularization, this property is retained after decomposition as well. Alas, answering this is beyond the scope of this paper. Luckily, however, in practice it is usually the opposite directionality that is of much more relevance and which can be established easily by inductive reasoning:

Theorem 1: Consider a network composed from multiple submodules. If all individual modules of this network converge, the overall model converges as well.

Intuitively, the meta-search space of the overall model is the superposition of the individual modules' sub-search spaces. Similarly, the optimum in the composite space is formed by superimposing the optima of the subspaces. Obviously, as long as all submodules converge against their respective optima, the overall model equally converges. If all subspaces are disjoint, the resulting meta-optimum is a point; otherwise it consists of a higher dimensional region.

Note, however, that the optimum found this way is not necessarily the global optimum of the meta-space. The problem of finding a decomposition that guarantees the latter is strongly related to the question posed at the beginning of this section and as such equally beyond the scope of this work.

Conversely, convergence within the individual submodules can be understood as convergence in the meta-space along the

dimensions of the respective subspace against the projection of the meta-optimum to this subspace. This is comparable to freezing all DOFs of the meta-space except for the ones of a specific submodule and solely performing optimization, e. g. via SGD (Stochastic Gradient Descent), on the remainder. Albeit the fineness of the descent consequently decreases, the convergence properties remain unaffected.

As an additional observation, the maximum improvement of performance a composed network can achieve over a comparative monolithic model is limited to the amount of information immanent to the modularization. If the modularization does not add any auxiliary information to the model, the original search space is retained and the global optimum remains the same in consequence as well. More specific decompositions infer more prior knowledge and hence the higher the theoretically possible performance gain; the model becomes more susceptible to modularization error at the same time, however (cf. section IV-D).

D. Modularization Error

A major challenge of modularization is finding a suitable decomposition that projects the prior knowledge concerning the structure of task or data base to the model without at the same time causing the loss of pivotal primary information e. g. due to oversimplification, etc. Noteworthily, the information is not really lost; the decomposition simply causes the network to lose the capability to model certain interrelations in this case due to the inferred constraints. Hence decomposition on the task and model layer can have adverse effects on the performance of the model if it decomposes the overall search space in a way that the theoretically optimal solution is located outside of the subspaces. We denote this as a suboptimal decomposition and the resulting deviation between theoretic optimum and closest reachable configuration within the composed search space as modularization error.

Furthermore, even if an optimal decomposition can be found, modularization may also affect the topology of the search space in a way that the inherent characteristics of optima finding algorithms – most commonly SGD – are affected negatively (e. g. by resulting in subspaces that are almost completely sparse). This may additionally impact efficiency and performance of the model.

Consequently, finding a suitable decomposition is a sensitive issue and should always be performed on a per-model basis for the performance loss due to modularization errors not to outweigh the practical advantages of the approach.

V. Measuring Model Trustworthiness

While we consider it to be self-evident that, through explicit structural definition and the availability of interim results, modularization simplifies interpretability and facilitates error traceability, the implications of the approach on model trustworthiness necessitate further investigation.

First and foremost, the question arises as to what constitutes trustworthiness in relation to neural networks. According to the "Ethics guidelines for trustworthy AI" by the European Commission ([13]), trustworthy AI must be lawful, ethical and robust. While the first lawfulness and ethicality are arguably abstract concepts, robustness is again subdivided into different technical aspects such as transparency and reliability. In this paper, we focus on the latter; in particular, we argue that a pivotal factor of reliability that is facilitated by modularization is the predictive conceptual proximity of a model, i. e. how close faulty predictions are on average to the ground truth on a conceptual level.

To elaborate this point, consider e. g. the obstacle detection system of an autonomous vehicle. In this case it is of less importance for such a system to be able to accurately tell apart *men* and *woman* than it is for it reliably distinguish between *humans* and e. g. *road signs*. This we denote as predictive conceptual proximity.

To measure this property, we employ semantic distance as defined by [2] as

$$S_{i,j} = \frac{\text{intersect}(\text{path}(i), \text{path}(j))}{\max(\text{length}(\text{path}(i)), \text{length}(\text{path}(j)))}$$

where $\text{path}(i)$ is the path from the root node to the i-th node in a hierarchical label structure as we use it in our experiments. I. e. the higher the semantic distance, the closer the two concepts are related. Measured over a set of n samples, it is hence possible to obtain an average semantic distance for a whole model as

$$\bar{S} = \frac{1}{n} \cdot \sum^{n} S_{predicted,ground\_truth}$$

This metric is, however, directly dependent on the absolute accuracy of the respective network. This makes it a poor choice when comparing two different models with differing performance levels. Consider e. g. two networks, one with a top-1 accuracy of 70% and an average semantic distance of 0.7 and one with a top-1 accuracy of 50% and a semantic distance of 0.65. Solely asking which of both has better learned categorical interrelations in the data base, the answer is obviously the second one; yet, on absolute terms, the first network is still better.

To reflect such differing performance levels when comparing models, we define the standardized semantic distance as the quotient of the average semantic distance and the categorical accuracy of the respective model

$$\bar{S}_{norm} = \frac{\bar{S}}{acc}$$

Hence models with higher conceptual proximity relative to their performance receive higher scores than networks with high accuracy but comparably low conceptual proximity.

A different interpretation of this metric would intuitively be that it measures to which extend a model utilizes the available resources to learning a direct mapping between inputs and outputs in comparison to learning categorical interrelations, i. e. how well it memorizes versus how well it generalizes.

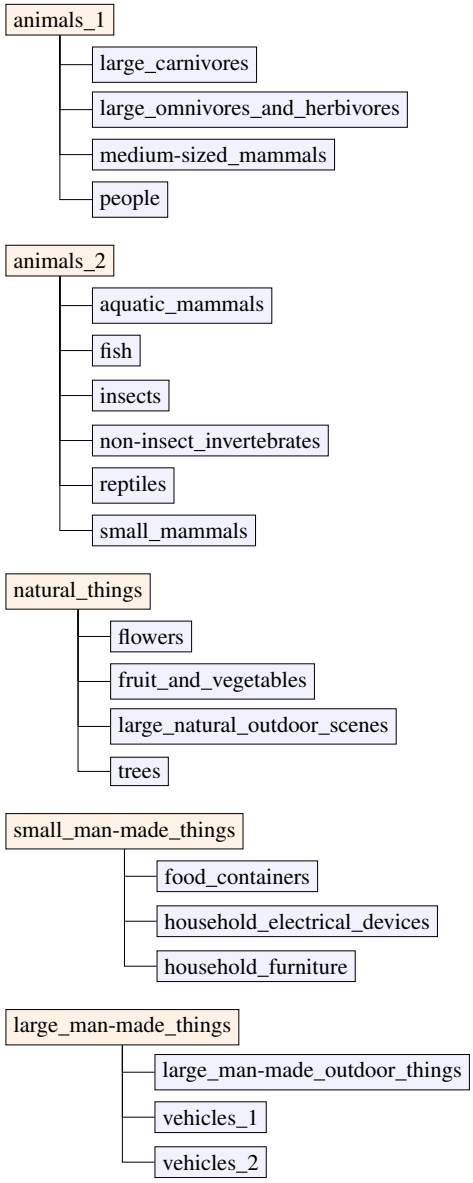

Fig. 3. Augmented label space of the CIFAR100 dataset; top-level categories (orange) were added manually, no changes were made to the fine label assignment per coarse category (fine-level labels omitted for the sake of legibility).

VI. Experiments

A. General

To assess the validity of the initially formulated hypotheses regarding the implications of modularization on model trustworthiness, we conducted two experiments on the task of image recognition. We used an augmented version of the CIFAR100 dataset [14] for which we manually extended the label space by a third hierarchy layer as depicted in figure 3 as the data base.

In both experiments, we compared performance and (standardized) semantic distance of a hierarchically composed network to a monolithic reference model. The same reference architecture was utilized for all models, specifically the MaxOut architecture as described by [15].

Similarly, the standard 10-crop procedure as described by [16] was - with minor modifications - employed for preprocessing, training and evaluation in each test case. Noteworthy, however, global contrast correction and Zero Component Analysis whitening for decorrelation of pixel values ([14], [15]) was applied to the dataset as a whole instead of random eigenvector-based alteration of the RGB channels' intensities of each image individually.

Predictions for the composite networks were calculated layer-wise. First, batches were fed to each module of a specific layer individually; afterwards, the predictions of each submodule were additively aggregated weighted by their respective scores of the preceding layer to form the final prediction for that layer.

It should be noted that, even though there are several promising approaches to be found in literature regarding possible improvements of the resilience of hierarchically composed models (cf. e. g. [2], [17] or [10]), we avoided incorporating these ideas into our experiments as the goal of this work is to examine whether modularization of networks in general is a feasible approach and we wanted to keep our results as unbiased as possible in this regard. Nevertheless, it should also be noted that in doing so, we assume that we leave lots of room for improvement.

B. Homogeneous Model Size

In the first experiment, number of trainable weights was used as a factor for the comparability of the models as we do assume that if two networks have roughly the same amount of DOFs they do possess a roughly equivalent modelling capability. Hence the composed model was designed to have roughly the same number of parameters as the reference model.

The meta-structure of the composed network adhered to that of the dataset. We trained a total of 26 modules, one module for the top layer and for each subcategory. Except for the number of outputs that differed based on the layer, all submodules were kept identical. Each model consisted of a total of three convolutional MaxOut layers with 32, 64 and 128 units each, kernel size 3x3, stride 1x1 and MaxOut resolution 4 (i. e. one MaxOut unit interpolates the superposition of eight linear functions). Each convolutional layer was connected to a MaxPool layer with pooling size and stride 2x2 and dimension preserving padding.

After flattening the output of the last convolutional layer it was forwarded to a fully connected MaxOut layer with 64 units and MaxOut resolution 4, followed by the softmax output layer with the number of units equal to the number of classes on the respective hierarchy level (i. e. 20 on the coarse and 100 on the fine layer). For weight regularization, a combination of Dropout preceding and subsequent to the fully connected layer with activation rate 0.5 and MaxNorm as [18] report to be most effective was employed. Furthermore, Dropout was also applied to the convolutional layers with an activation rate

TABLE I
EXPERIMENTAL RESULTS

Metric	Comp. Net. 1	Comp. Net. 2	Benchmark
Cat. Acc. Top	71.12%	**78.16%**	78.1%
Cat. Acc. Coarse	50.17%	59.03%	**62.49%**
Cat. Acc. Fine	34.28%	39.31%	**50.42%**
Sem. Dist. (avg.)	0.52	0.58	**0.63**
Sem. Dist. (std.)	**1.55**	1.51	1.26

of 0.1 to facilitate the discovery of informative features as described in [19]. All modules combined, the composed model possessed a total of 51,619,880 trainable weights.

The structure of the benchmark model followed the same basic architecture with the difference that the convolutional layers consisted of 128, 256 and 512 and the fully connected layer of 256 units respectively and a MaxOut resolution of 8 was used at all layers, making for a total of 49,369,988 trainable parameters.

For training, the original CIFAR100 training set was split up into a training and a validation set at a ratio of 4:1, i. e. 40,000 and 10,000 records respectively. As stated before, preprocessing and augmentation of both datasets was done in adherence to a slightly modified version of the standard 10-crop procedure as described by [16]. Training was done using categorical crossentropy as the loss function. Weight updates were performed using Adam (cf. [20]) with $\beta_1 = 0.9$, $\beta_2 = 0.999$ and $\epsilon = 10^{-7}$ as the optimizer. The learning rate was set dynamically depending on the training epoch, beginning with an initial rate of 10^{-3} which was lowered to 10^{-4} after the third and to 10^{-5} after the sixth epoch for the submodules of the composed network resp. 10^{-4}, 10^{-5} and 10^{-6} for the reference model. To avoid overfitting, early stopping with a minimum loss delta on the validation set of 10^{-2} and a patience of three epochs was employed during the whole training. Weight updates were performed until either convergence of the validation loss.

The reported hyperparameters in both cases were reached by starting off from sensible initial values and continuously improving them through empirical testing over several test runs until the results were in roughly the same order of magnitude as those reported by [15]. It should be noted though that no excessive optimization was performed.

During testing, the categorical accuracy on each hierarchy level as well as the standardized semantic distance as described in section V were measured for both networks. The results are depicted in table I.

We achieved a top-1 categorical accuracy of 34.28% and 50.42% with the composed resp. the reference model. Likewise, a categorical accuracy of 50.17% and 62.49% could be observed on the coarse layer (i. e. top-5 accuracy) and 71.12% resp. 78.1% on the top layer. As expected, the composed network performed worse than the monolithic model under these constraints. Moreover, it can be observed that the intuitive expectation expressed in the form of the modularization error, that the error increases steadily with increasing hierarchy layer, is confirmed.

Similar to the accuracy, the average semantic distance of the composed network is lower with 0.52 in comparison to 0.63 as well; however, with regard to the standardized semantic distance, the former easily outperforms the reference model with 1.55 versus 1.26. Hence, even though the latter makes fewer errors, its wrong decisions are relatively further off than those of the composite network. In other words, by learning higher-level concepts, the predictions of the latter are on average contextually closer to the ground truth. As such, it can be concluded that even with this fairly naive decomposition, the resulting model is more trustworthy than a comparative monolithic model.

An additional qualitative – albeit intuitive – observation that became apparent during the experiment is that the reduced size of the individual submodules of the composite network significantly reduced their training time relative to the reference model, making the tuning process much more flexible.

C. Layer-wise Size Adaption

In the first experiment, it was shown that modularization can indeed have a positive effect on trustworthiness with respect to the conceptual proximity of predictions of neural networks. However, due to the strict requirements with regard to the model architecture in combination with a relatively naive decomposition, the absolute performance was significantly worse at the same time. Thus, in the second experiment, we relaxed the constraints for the composite network in such that removed the requirement that all submodules together must possess the same amount of trainable weights as the reference model to examine whether it is possible to achieve the desirable impact on the predictive behavior while at the same time retaining a comparable degree of performance.

The first adjustment we made was to continuously decrease the submodule size of the composite network with progressing hierarchical levels. We hypothesized that, due to the modularization error, the performance of the earlier levels is relatively more important for the conceptual proximity of the model's predictions than that of the more downstream levels. In practice, this can be translated intuitively to the extent that it is more important for the trustworthiness of a network that a reliable distinction is made between e. g. *tree* and *insect* than between *woman* and *girl* and that resources should hence be allocated accordingly. Thus we increased the number of units per convolutional MaxOut layer to 64, 128 and 256 for the coarse and the top level modules; in the case of the latter, we furthermore doubled the number of units in the fully connected layer.

Moreover, we speculated that too small filter sizes in the convolutional layers would prove counterproductive the more general the concepts to be learned are. Therefore we increased the filter size for the coarse layer to 4x4 and for the top layer to 5x5.

Everything else being kept the same, with these two changes, we achieve a top-1 categorical accuracy of 39.31%. On the coarse layer, a categorical accuracy of 59.03% could be observed; on the top layer 78.16%. The average semantic

distance increases to 0.58 while the standardized semantic distance slightly deteriorates to 1.51. The latter indicates that the performance gain in comparison to the first experiment is mostly attributable to the direct memorization of classes; in other words, the model utilizes the additional degrees of freedom to a greater extent to learn the mapping between input data and outputs than to learn the relations between the different categories. Hence the ratio between average semantic distance and the absolute categorical accuracy decreases.

Regardless, it can be seen that with only two very simple adaptions the performance of the modularized network can be greatly enhanced so that the top-layer accuracy is equal to and the top-5 accuracy is only slightly behind the reference model while the predictive conceptual proximity measured in the standardized semantic distance is constant. Especially considering that module architecture is but one of several optimization vectors, most of which we don't take into account in this work (such as the modularization structure which we simply adopt from the label space of the dataset here), we do hence reason that it seems indeed possible to achieve the desirable increase in conceptual proximity while retaining a comparable degree of performance through modularization.

VII. Conclusion

We proposed modularization as a technique to address several inherent shortcomings of DL and to facilitate interpretability and trustworthiness of neural networks, particularly CNNs. Noteworthily, we consider our approach to be orthogonal (i. e. not obtrusive) to conventional performance-oriented methods.

Advantages of the proposed technique – namely increased transparency and error traceability, enhanced retrainability and modifiability of models, simplified refactoring of once-learnt knowledge and reduced hardware requirements – were discussed at a conceptual level. A theoretical foundation with regard to decomposable dimensions of ML models in general, convergence and performance properties of composed networks and different types of modularization was established. Moreover, the concept of modularization errors was introduced and a heuristic to estimate the impact of a specific form of decomposition on the performance of a model was provided.

Exemplary experiments were conducted on the CIFAR100 dataset to assess the technique's qualitative impact it has on the model's behavior. Specifically, the overall performance and the conceptual relatedness between predictions and ground truths as a measure of model trustworthiness were examined each for a composite and a monolithic reference network.

The results suggest that (sensible) modularization indeed incentivizes networks to learn categorical interrelationships better than without explicit structural definition, resulting in conceptually closer predictions. As such we conclude that the presented approach facilitates the trustworthiness of models. Generally, a decrease in performance proportional to the quality of the decomposition could be observed as a trade-off for desirable characteristics such as an simplified interpretability and better error traceability for the composite networks on the lower hierarchy layers of the composite networks.

In conclusion, we do believe the presented approach possesses great potential for many practical applications. There is, however, still lots of room for further research with regard to modularization.

References

[1] M. Z. Alom, T. M. Taha, C. Yakopcic, S. Westberg, P. Sidike, M. S. Nasrin, B. C. Van Esesn, A. A. S. Awwal, and V. K. Asari, "The history began from AlexNet: A comprehensive survey on deep learning approaches," Mar. 2018.

[2] R. Fergus, H. Bernal, Y. Weiss, and A. Torralba, "Semantic label sharing for learning with many categories," pp. 762–775, 2010.

[3] R. A. Jacobs, M. I. Jordan, and A. G. Barto, "Task decomposition through competition in a modular connectionist architecture: The what and where vision tasks," *Cognitive science*, vol. 15, no. 2, pp. 219–250, Apr. 1991.

[4] R. A. Jacobs, M. I. Jordan, S. J. Nowlan, and G. E. Hinton, "Adaptive mixtures of local experts," *Neural computation*, vol. 3, no. 1, pp. 79–87, 1991.

[5] M. I. Jordan and R. A. Jacobs, "Hierarchical mixtures of experts and the EM algorithm," *Neural computation*, vol. 6, no. 2, pp. 181–214, Mar. 1994.

[6] J. A. Benediktsson and P. H. Swain, "Consensus theoretic classification methods," *IEEE transactions on systems, man, and cybernetics*, vol. 22, no. 4, pp. 688–704, Aug. 1992.

[7] L. Xu, A. Krzyzak, and C. Y. Suen, "Methods of combining multiple classifiers and their applications to handwriting recognition," *IEEE transactions on systems, man, and cybernetics*, vol. 22, no. 3, pp. 418–435, May 1992.

[8] E. Alpaydin, *Maschinelles Lernen*, 2nd ed. Walter de Gruyter GmbH & Co KG, May 2019.

[9] Z. Yan, H. Zhang, R. Piramuthu, V. Jagadeesh, D. DeCoste, W. Di, and Y. Yu, "HD-CNN: Hierarchical deep convolutional neural networks for large scale visual recognition," 2015.

[10] D. Roy, P. Panda, and K. Roy, "Tree-CNN: A hierarchical deep convolutional neural network for incremental learning," *Neural networks: the official journal of the International Neural Network Society*, vol. 121, pp. 148–160, Jan. 2020.

[11] D. Filan, S. Hod, C. Wild, A. Critch, and S. Russell, "Neural networks are surprisingly modular," Mar. 2020.

[12] D. H. Wolpert and W. G. Macready, "No free lunch theorems for optimization," *IEEE Transactions on Evolutionary Computation*, vol. 1, no. 1, pp. 67–82, Apr. 1997.

[13] Robotics and Artificial Intelligence (Unit A. 1), "Ethics guidelines for trustworthy AI," https://ec.europa.eu/digital-single-market/en/news/ethics-guidelines-trustworthy-ai, Apr. 2019, accessed: 2020-8-1.

[14] A. Krizhevsky, "Learning multiple layers of features from tiny images," *Master's thesis, University of Tront*, 2009.

[15] I. J. Goodfellow, D. Warde-Farley, M. Mirza, A. Courville, and Y. Bengio, "Maxout networks," Feb. 2013.

[16] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "ImageNet classification with deep convolutional neural networks," in *Advances in Neural Information Processing Systems 25*, F. Pereira, C. J. C. Burges, L. Bottou, and K. Q. Weinberger, Eds. Curran Associates, Inc., 2012, vol. 25, pp. 1097–1105.

[17] J. Deng, N. Ding, Y. Jia, A. Frome, K. Murphy, S. Bengio, Y. Li, H. Neven, and H. Adam, "Large-Scale object classification using label relation graphs," pp. 48–64, 2014.

[18] N. Srivastava, G. Hinton, A. Krizhevsky, I. Sutskever, and R. Salakhutdinov, "Dropout: A simple way to prevent neural networks from overfitting," *Journal of Machine Learning Research*, vol. 15, pp. 1929–1958, Jun. 2014.

[19] S. Park and N. Kwak, "Analysis on the dropout effect in convolutional neural networks," in *Computer Vision – ACCV 2016: 13th Asian Conference on Computer Vision, Taipei, Taiwan, November 20-24, 2016, Revised Selected Papers, Part II*, ser. Lecture Notes in Computer Science, S.-H. Lai, V. Lepetit, K. Nishino, and Y. Sato, Eds. Cham: Springer International Publishing, 2017, vol. 10112, pp. 189–204.

[20] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," in *3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings*, Y. Bengio and Y. LeCun, Eds., 2015.