OBJECT DETECTION WITH OOD GENERALIZABLE NEURAL ARCHITECTURE SEARCH

Anonymous authors
Paper under double-blind review

ABSTRACT

We present a Neural Architecture Search (NAS) framework guided by feature orthogonalization to improve Out-of-Distribution (OOD) Generalization on Object Detection. Specifically, we attribute the failure of generalizing on OOD data to the spurious correlations of category-related features and context-related features. The category-related features describe the causal information for predicting the target objects, e.g., "a **car** with four **wheels**", while the context-related features describe the non-causal information, e.g., "a **car** driving at **night**", and the context-related features are always mistaken for causal information due to the existence of distinct data distribution between training and testing sets (OOD) to some degree. Therefore, we aim at automatically discovering an optimal architecture that is able to disentangle the category-related features and the context-related features with a novel weight-based detector head. Both theoretical and experimental results show that the proposed scheme is able to achieve the disentanglement and better performance on both Independent-Identically-Distribution datasets (Pascal VOC 2012 and MS COCO) and OOD datasets (BDD100K-weather and BDD100K-time-of-day).

1 INTRODUCTION

Object detection is a fundamental task in computer vision. However, the generalization ability of object detection remains a challenging problem, especially for Out-of-Distribution (OOD) scenarios, where data are sampled from novel unseen distributions. For example, imagine the following situation: a self-driving car equipped with an object detection system to detect cars and pedestrians on the roads. The performance of the object detection system can drop significantly when facing OOD scenarios, for example, new city or weather scenes that do not exist in the training set. This may lead to serious accidents as shown in worldwide news about self-driving car accidents that usually happen on scenes rarely seen in training set (Law, 2021).

Among all efforts to improve the generalization ability of object detectors, neural architecture search (NAS) methods have been proven to be an effective way when facing the Independent-Identically-Distribution (IID) datasets. However, these methods' performance may suffer from severe performance degradation when facing OOD data due to the easily over-fitting nature of NAS methods (Chen et al., 2019b; Cai et al., 2018; Jiang et al., 2020; Guo et al., 2020). On the other hand, recent OOD challenges have sprung a series of works on improving the OOD generalization abilities of deep neural networks (DNNs) (Bahng et al., 2020; Bai et al., 2020; Krueger et al., 2021a). These works can be categorized into invariant risk regularization methods (Arjovsky et al., 2019; Ahuja et al., 2020), domain generalization (DG) methods (Carlucci et al., 2019; Li et al., 2017; Dou et al., 2019; Li et al., 2021; Krueger et al., 2021b; Pezeshki et al., 2020; Sagawa et al., 2019; Koyama & Yamaguchi, 2021), and disentangled representation methods (Liu et al., 2018b; Peng et al., 2019). However, these works only show moderate performance improvement compared with the standard empirical risk minimization, when evaluated on more practical datasets (Gulrajani & Lopez-Paz, 2020; Ye et al., 2021), which are reliant on utopian hypotheses specifically designed for image classification tasks only.

In this work, we focus on the OOD object detection task for improving the unseen domain performance of object detectors trained on limited data distributions to generalize to different data distributions. This is achieved by a differentiable NAS search for disentangling the extracted feature into the category-related branch and the context-related branch via feature orthogonalization, where category

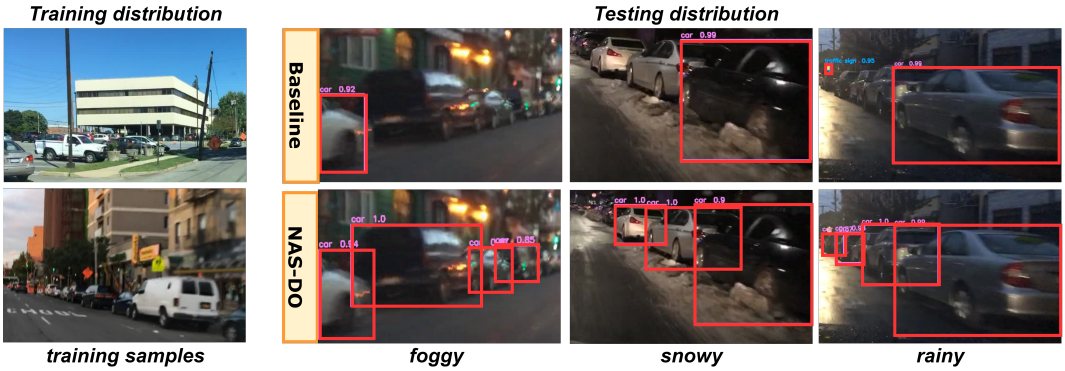


Figure 1: OOD object detection aims at generalizing to the unseen testing distribution based on the training distribution. Some evaluation examples of our proposed methods and baseline (SwinTransformer (Liu et al., 2021)).

information and context information are respectively captured by these two branches. Experimental results show that our proposed head derives robust architecture and performs well in extreme OOD environments (some examples in Figure 1), where there is a huge data distribution gap between the training set and testing set. Our main contributions can be summarized as followed:

- We systematically analyze the performance improvement of existing OOD generalization algorithms for object detection and demonstrate that most of them are not effective.
- We propose a novel differentiable neural architecture search framework on the backbone network for object detection, namely NAS-DO, guided by feature orthogonalization to disentangle the causal information for object detection and the non-causal information. The proposed algorithmic framework has achieved the best performances on challenging OOD scenarios with up to 20% improvement compared to baselines.
- We theoretically prove the effectiveness of feature orthogonalization constraint for category and context feature disentanglement as well as the convergence of the proposed algorithm.

2 Related Work

2.1 NAS on Object Detection

Compared with NAS works for the standard image classification tasks, the works of NAS for Object Detection are relatively rare due to their intricacy. Existing works on NAS for Object Detection can be generally divided into three genres according to the searched component in networks, including backbone search, feature pyramid (FPN) network search, and joint detection head and FPN search. For the backbone search type, Chen *et al*. searches for an efficient backbone by applying single-path training to reduce approximation bias of super-net (Chen et al., 2019b) following (Cai et al., 2018; Guo et al., 2020). Jiang *et al*. further improved it with a serial-to-parallel backbone searching strategy (Jiang et al., 2020) to properly allocate the computation and better fuse high-level features into low-level features (Wang et al., 2020a). For feature pyramid network search, Ghiasi *et al*. designed a search space of scalable architecture to generate multi-scale feature representations (Ghiasi et al., 2019). Liang *et al*. searches for efficient and more adaptive FPN from the pre-trained super-net by proposing a one-shot NAS framework (Liang et al., 2021). The third one is the joint FPN and detection head search. Xu *et al*. focuses on improving the feature fusion and detection head modules to discover a task-specific network that can adapt well to any dataset (Xu et al., 2019). NAS-FCOS aims to efficiently search for FPN as well as the prediction head by using a reinforcement learning paradigm (Wang et al., 2020b). The existing NAS methods for object detection mainly focus on IID setting and this limitation usually leads to over-fitting since the training set and the testing set are derived from the same distribution, which motivates us to consider OOD generalizable NAS. Bai et al. (2021) have developed a differentiable NAS framework for OOD generalization classification with a conditional generator, however, it is generally hard to train the conditional generator for object detection, as images usually involve more than one objects.

2.2 Out-of-Distribution Generalization

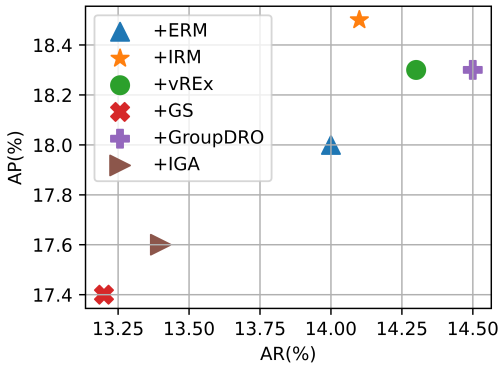


Figure 2: Results of Faster R-CNN (Ren et al., 2015) with ResNet-50 (He et al., 2016) backbone +ERM (Vapnik, 1998), +IRM (Arjovsky et al., 2019), +vREx (Krueger et al., 2021b), +GS (Pezeshki et al., 2020), +GroupDRO (Sagawa et al., 2019), +IGA (Koyama & Yamaguchi, 2021) on BDD100K-weather.

Out-of-Distribution (OOD) Generalization, the task of generalizing under such data distribution shifts, has raised broad interest recently. These works can be grouped into these categories, including the domain generalization (Peng et al., 2019; Bai et al., 2020; Dou et al., 2019; Ganin et al., 2016), the causal inference methods (Peters et al., 2017), and the invariant learning methods (Arjovsky et al., 2019; Ahuja et al., 2020). For example, Peng *et al.* (Peng et al., 2019) devise an auto-encoder model to disentangle domain-specific features from class identity. Dou *et al.* (Dou et al., 2019) improves the generalization performance by aligning a derived confusion matrix of classification with preserved general knowledge prior to inter-class relationships. Motivated by learning the invariance from the heterogeneity that existed in data for classification, the invariant risk minimization method achieves OOD generalization by regularizing the classifier to achieve similar performance across different subsets of datasets (Arjovsky et al., 2019). Ahuja *et al.* further improve its stability due to the strong regularization effects in optimization (Ahuja et al., 2020). However, it is not easy to directly apply these methods for object detection tasks. For example, many domain generalization algorithms rely on special structures for classification, such as mixup (Yan et al., 2020), which is not applicable for object detection tasks as the inputs will be demolished by mixing objects with backgrounds and the bounding boxes will be chaos. Besides, other methods without introducing classification-specific structures, such as IRM (Arjovsky et al., 2019), introduce strong regularization effects that hinder the optimization process for complex object detection tasks, which may even lead to performance degeneration, as demonstrated in Figure 2. All of these indicate that it is a non-trivial challenge to achieve OOD generalization for object detection. Pham et al. (2021) have proved that generalizing to unseen testing distributions requires large models.

3 Methodology

3.1 Preliminaries on Differentiable Neural Architecture Search

Conventional differentiable neural architecture search methods utilize a gradient-based optimization to search the optimal sub-architecture (cell) of the super-net (Liu et al., 2018a; Yang et al., 2020). The super-net is mainly stacked by several cells which are the computation units to be searched during the training process. It can be represented by a directed acyclic graph (DAG). There are two types of cells, including the normal cell and the reduction cell which down-samples the feature map. A cell consists of n ordered nodes $X = \{x_1, x_2, \ldots, x_n\}$ and edges between nodes $E = \{e^{(i,j)} | 1 \leq i < j \leq n\}$. The output of each edge is the concatenation of m candidate operations $O = \{o_1, o_2, \ldots, o_m\}$. Binary variables $\alpha_k^{(i,j)} \in \{0, 1\}$ represent which operation(s) will be active. Thus, we have the following formulations for each node:

$$x_j = \sum_{i=1}^{j-1} \sum_{k=1}^{m} \alpha_k^{(i,j)} o_k(x_i) = \boldsymbol{\alpha}_j^T \mathbf{o}_j \tag{1}$$

where $\boldsymbol{\alpha}_j^T$ and $\mathbf{o}_j$ are vectors formed by $\alpha_k^{(i,j)}$ and $o_k(x_i)$ respectively. Since it is hard to optimize discrete value in a differentiable manner, DARTS-based (Liu et al., 2018a) methods convert $\alpha_k^{(i,j)}$ into continuous relaxation with a $softmax$ function:

$$s_k^{(i,j)} = exp(\alpha_k^{(i,j)}) / \sum_k exp(\alpha_k^{(i,j)}) \tag{2}$$

$$x_j = \sum_{i=1}^{j-1} \sum_{k=1}^{m} s_k^{(i,j)} o_k(x_i) = \mathbf{s}_j^T \mathbf{o}_j \tag{3}$$

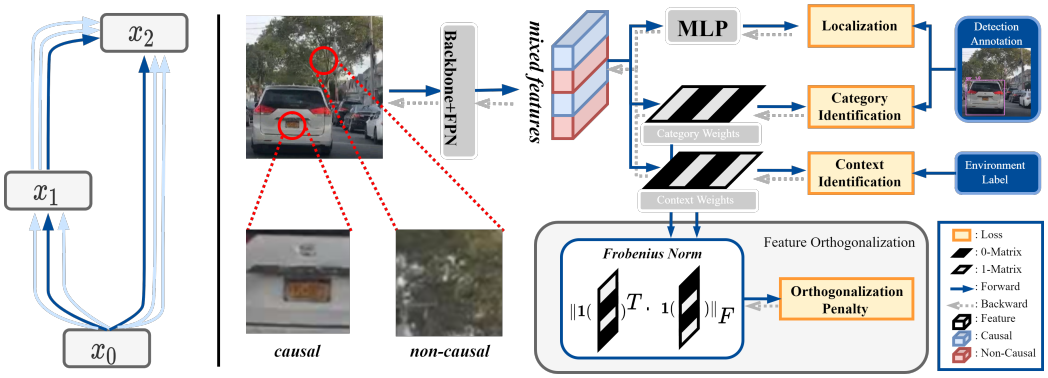


Figure 3: Overview of the NAS-DO mainstream. The searching backbone is stacked by normal cells and reduction cells, where normal cells have wiser channel output. Both normal and reduction cells comprise several ordered nodes and each edge between pair of nodes represents the weighted sum of candidate operations. Weights of category and context branch are orthogonal, therefore, the features extracted by these branches are orthogonal as well.

$s_k^{(i,j)}$ are trainable parameters and the problem is formulated as the following bi-level optimization problem:

$$\mathbf{s}^* = \arg\min_s \mathcal{L}_{val}(\boldsymbol{\omega}^*, \mathbf{s}) \tag{4}$$

$$\boldsymbol{\omega}^* = \arg\min_{\omega} \mathcal{L}_{train}(\boldsymbol{\omega}, \mathbf{s}) \tag{5}$$

$$s.t.\ \|\mathbf{s}_j\|_0 = a, 1 \leq j \leq n \tag{6}$$

where $\mathbf{s}$ and $\boldsymbol{\omega}$ denote architecture parameters and network weights respectively. $\mathcal{L}_{val}$ denotes the validation loss and $\mathcal{L}_{train}$ denotes the training loss. Constant a is the sparseness, *i.e.*, $a = 2$ indicates keeping the top-2 strongest dimensions for node j. During searching process, $\mathcal{L}_{val}$ and $\mathcal{L}_{train}$ are optimized alternately (Liu et al., 2018a). However, there is an inconsistency between high-performance super-net and target-net caused by the two-stage methods. Inspired by (Yang et al., 2020), we apply a one-stage manner with the architecture parameters constraint satisfied by formulating new architectures generating problem as a sparse coding problem to eliminate this performance gap:

$$z_j = \arg\min_z \frac{1}{2}\|A_j z - s_j\|_2^2 + \lambda\|z\|_1, 1 \leq j \leq n \tag{7}$$

where $A_j \in \mathbb{R}^{p_j \times (j-1)m}, p_j \leq (j-1)m$ denotes the measurement matrix, $s_j \in \mathbb{R}^{(j-1)m}$ are architecture parameters and z_j is the sparse signal. The sub-net $N_{S(z)}$ of the super-net is derived from the support set $S(z)$ which is projected by z_j.

3.2 Search Space Design

Normal cells and reduction cells are the smallest searched units and the whole searching space is alternately stacked by these two types of cells. We extract the output of the last four cells as the input of the feature pyramid network followed by detector heads to predict locations and categories. Moreover, inspired by the success of the attention mechanism (Vaswani et al., 2017), we construct the searching cells with two types of attention layers and the definitions of candidate operations $O = \{o_1, o_2, \ldots, o_m\}$ are listed as follow:

Attention_layer_sparse(op_0). Arguments include C_{in}(input channel), C_{out}(output channel), $kernel\_size$, $stride$ and $padding$. The whole structure contains two sub-structures, the first one is the basic layer (Liu et al., 2021) and the other is the convolution block which is applied to maintain the channel of input and output tensor to be consistent with C_{in} and C_{out}. We set the dimension to 96, depth to 2 and head number to 2 for the basic layer.

Attention_layer_dense(op_1). The difference between op_0 and op_1 is that op_1 is deeper and wider than op_0 with 192 dimensions, 4 depth and 4 head number for basic layer.

Algorithm 1: Object Detection with OOD Generalizable Neural Architecture Search

Input: training set $\mathcal{D}$, batch size n, learning rate β, searching .
Output: An architecture with optimized parameters.
Initialize super-net $\mathcal{N}(\boldsymbol{\omega}, \mathbf{s})$; $search\_flag \leftarrow True$.
while $not\ converged$ **do**
  if $search\_flag$ **then**
    Recover z by solving Eq. 7 and project the support set $S(z) = \{i|z(i) \neq 0\}$.
    Derive the sub-net $N_{S(z)}$; $z_{new} := z$.
    if $\|z_{new} - z_{old}\| \leq \epsilon$ **then**
      $search\_flag \leftarrow False$.
    end if
  end if
  for $enumerate\ train\ set$ **do**
    Sample a batch of data $\{(x_i, y_i, y\_ctx_i)\}_{i=1}^{n}$.
    Calculate $\mathcal{L}_{train}$ according to Eq. 10.
    $\boldsymbol{\omega} \leftarrow \boldsymbol{\omega} - \beta \cdot \nabla \mathcal{L}_{train}(\mathcal{N}_{S(z)}(\boldsymbol{\omega}, \mathbf{s}))$.
    $\mathbf{s} \leftarrow \mathbf{s} - \beta \cdot \nabla \mathcal{L}_{train}(\mathcal{N}_{S(z)}(\boldsymbol{\omega}, \mathbf{s}))$.
  end for
  $z_{old} := z_{new}$.
end while

Skip_connect(op_2) (Melis et al., 2017). If the current cell is a normal cell, then the size of the output is the same as the input. If the current cell is a reduction cell, we use a convolutional layer with C_{in} input channels and C_{out} output channels to maintain consistency.

3.3 ALGORITHM FRAMEWORK

Our searching process is outlined in Algorithm 1 and the overview of NAS-DO is visualized in Figure 3. Firstly, a super-net backbone and heads are constructed for search. Then, we initialize the super-net parameters, including network weights $\boldsymbol{\omega}$ and architecture parameters $\mathbf{s}$. To control the searching loop, we use a termination condition when the z of two neighbor iterations are closed. z is recovered by solving the sparse coding problem (Eq. 7) and then derive the sparse sub-net $N_{S(z)}$. Lastly, network weights $\boldsymbol{\omega}$ and architecture parameters $\mathbf{s}$ are optimized by descending gradients using training loss.

Feature orthogonalization. To disentangle the extracted features, we design a two-branch detector head (see the category and context identification in Figure 3), which is comprised of two classifiers to predict category label and context label respectively and impose weight-based loss to constrain the category branch weight W_{cls} and context branch weight W_{ctx} to be orthogonal using context labels [1]:

$$\mathcal{L}_{feat\_orth} = \|\mathbb{1}(W_{cls})^T \mathbb{1}(W_{ctx})\|_F \tag{8}$$

where $\mathbb{1}(x)$ is the element-wise indicator function, $\mathbb{1}(x) = 1$, if $x \neq 0$, otherwise, $\mathbb{1}(x) = 0$. $\|\cdot\|_F$ is *Frobenius Norm*.

Overall loss. For the context branch, we adopt the same loss function as the category branch using image context labels:

$$\mathcal{L}_{ctx} = CE(Y_{ctx}(X), Y^*_{ctx}(X)) \tag{9}$$

where CE refers to cross-entropy loss function; Y_{ctx}, Y^*_{ctx} indicates the ground-truth context labels and output context labels respectively. Thus, the overall training loss is defined as:

$$\mathcal{L}_{train} = \mathcal{L}_{cls} + \mathcal{L}_{reg} + \lambda_{ctx} \cdot \mathcal{L}_{ctx} + \lambda_p \cdot \mathcal{L}_{feat\_orth} \tag{10}$$

where $\mathcal{L}_{cls}$ and $\mathcal{L}_{reg}$ are consistent with (Cai & Vasconcelos, 2018), λ_{ctx} and λ_p are hyper-parameters.

[1] The context labels are actually the domain labels which indicate the domain where images are drawn from, and using such labels is a very common practice in Domain Generalization researches (Section 2.2)

Table 1: IID results comparisons on **Pascal VOC2007 testing set** and **MS COCO val-set**. All models are trained from **scratch**. - indicates failures. R-50 and R-101 represent ResNet-50 and ResNet-101 backbone respectively. X-101 represents ResNeXt-101. All models share 256 FPN width. For baselines, we use models implemented by mmdetection (Chen et al., 2019a) and for Swin Transformer, we use official implementation provided by authors. *Note that NAS-DO has two parameter sizes for the two datasets respectively.*

MODEL	BACKBONE	PARAMS(M)	PASCAL VOC			MS COCO		
			AP	AP_{50}	AP_{75}	AP	AP_{50}	AP_{75}
CASCADE RCNN	X-101	127	5.5	13.2	3.5	-	-	-
RETINANET		94	21.2	39.5	19.8	-	-	-
NAS-FPN		116	31.5	53.8	32.1	13.3	22.8	13.4
NAS-FCOS		96	10.5	21.4	9.1	2.0	4.8	1.4
CASCADE RCNN	R-50	69	2.1	5.7	1.0	-	-	-
RETINANET		37	16.9	33.5	14.7	-	-	-
NAS-FPN		59	41.7	64.8	44.4	11.5	20.7	11.3
NAS-FCOS		38	10.7	23.5	8.5	1.8	4.4	1.3
CASCADE RCNN	R-101	88	2.4	6.6	1.2	-	-	-
RETINANET		55	16.0	32.1	13.9	-	-	-
NAS-FPN		78	40.9	63.9	43.3	12.2	21.3	12.1
NAS-FCOS		57	10.4	22.8	8.3	1.8	4.5	1.2
SWIN-T	SWIN	86	31.9	54.8	32.6	9.0	16.7	8.6
SWIN-S		107	45.7	69.8	49.4	11.9	21.5	11.6
SWIN-B		145	45.6	69.5	49.2	13.2	23.2	13.2
NAS-DO	NAS SPACE	153 & 143	**46.6**	**70.9**	**49.6**	**17.3**	**28.1**	**18.0**

3.4 THEORETICAL ANALYSIS

Feature orthogonalization for object detection

Considering in real practice, the category-related features are independent of the context, e.g., wheels of a car are not causal to the weather, thus, we have the following assumption:

Assumption 3.1. The category features B_{cls} and the context features B_{ctx} are independent $B_{cls} \perp\!\!\!\perp B_{ctx}$, and B_{cls} is independent to the context label Y_{ctx}, that is $B_{cls} \perp\!\!\!\perp Y_{ctx}$.

Intuitively, it is reasonable that the extracted features can be disentangled into causal and non-causal features, which indicates that the features can be written as a combination of category-related features and context-related features, then we have the following assumption:

Assumption 3.2. The input of the classifiers can be written as a concatenation (i.e. $X_C = [X_{C,cls}^T, X_{C,ctx}^T]^T$), where $X_{C,cls}$ is a function of the hidden category feature B_{cls}, (i.e. $\exists f_{cls} : \mathcal{R}^{B,cls} \to \mathcal{R}^{N_{C,cls}}$, $X_{C,cls} = f_{cls}(B_{cls})$), and $X_{C,ctx}$ is a function of the hidden context feature B_{ctx}, (i.e. $\exists f_{ctx} : \mathcal{R}^{B,ctx} \to \mathcal{R}^{N_{C,ctx}}$, $X_{C,ctx} = f_{ctx}(B_{ctx})$).

Constraint 3.3. The weights of the category and context classifiers are orthogonal, that is

$$\mathbb{1}(W_{cls})^T \mathbb{1}(W_{ctx}) = \mathbf{0} \tag{11}$$

Theorem 3.4. *(1) Theorem 3.1 and Theorem 3.2 hold; (2) the activation function is Lipschitz continuous; (3) the derivatives of the loss corresponding to the classifier outputs $Y_{C,cls}$ and $Y_{C,ctx}$, and the derivative of the activation function are stochastically bounded during the training; (4) the network width goes to infinity; (5) the sample size goes to infinity. Then, Theorem 3.3 is a sufficient condition for $Y_{C,cls} \perp\!\!\!\perp Y_{ctx}$.*

We prove Theorem 3.4 by using NTK (Neural Tangent Kernel) theorem, where conditions (2) to (4) are the conditions of NTK and are consistent with the conditions in (Jacot et al., 2018). Condition (5) guarantees the empirical distribution is close to the real distribution according to the Law of Large Number. Proof can be found in Appendix A.1.1.

Convergence of neural architecture search

Theorem 3.5. *Let $\mathcal{L}_{train}(\boldsymbol{\omega}, \boldsymbol{s})$ be continuous on $\boldsymbol{s}$ and $max\ \mathcal{L}_{train} \leq \infty$, then the sequence $\{z\}$ generated by Alg. 1 has limited points.*

Table 2: OOD results comparisons on BDD100K-weather and BDD100K-time-of-day. All models are trained from **scratch**. NAS-DO @ a-b-c represent the hyper-parameters of the searching space, which are a layers, b steps, c sparseness. AP_L and AR_L represent the average precision and average recall for objects with area $> 96^2$, where the area is measured as the number of pixels in the segmentation mask.

MODEL	BACKBONE	PARAMS(M)	WEATHER		TIME OF DAY	
			AP_L	AR_L	AP_L	AR_L
CASCADE RCNN	RESNEXT-101	127	3.1	5.5	1.5	5.3
RETINANET		94	4.0	10.4	3.6	11.5
NAS-FPN		116	15.1	28.2	14.3	24.8
NAS-FCOS		96	9.1	23.4	8.4	20.0
CASCADE RCNN	RESNET-101	88	1.2	3.2	0.4	3.3
RETINANET		55	5.1	13.4	4.6	12.0
NAS-FPN		78	24.6	33.9	23.7	33.4
NAS-FCOS		57	5.8	17.1	5.3	16.7
SWIN-S	SWIN	107	32.1	44.4	32.6	50.1
SWIN-B		145	33.5	46.0	34.7	51.8
NAS-DO (OURS) @ 4-4-2	$\lambda_{ctx} = 0$	109 & 152	50.4	58.1	36.5	44.8
NAS-DO (OURS) @ 4-2-2	$\lambda_{ctx} = 0.5$	101 & 109	51.1	59.1	37.4	46.8
NAS-DO (OURS) @ 4-4-2	$\lambda_{ctx} = 0.5$	166 & 150	**52.9**	**59.8**	**39.8**	**54.9**
COMPARE TO SWIN-B			**+19.4**	**+13.8**	**+5.1**	**+3.1**

The convergence of NAS-DO can be guaranteed in Theorem 3.5. Proof can be found in Appendix A.1.2.

4 EXPERIMENTS

In this section, we conduct numerical experiments to evaluate the effectiveness of NAS-DO on Pascal VOC (Everingham et al., 2010) and MS COCO (Lin et al., 2014) for standard IID performance evaluation and on BDD100K (Yu et al., 2018) for OOD scenarios. For the ablation study, we compare different search hyper-parameters of our search space and the weight of feature orthogonalization to find a balance between OOD generalization performance and algorithm complexity. Finally, we display the discovered architectures and some of the inference results in Appendix A.5. We also visualize the converged weights of the two-branch to illustrate the feature disentanglement during the optimization process.

4.1 IMPLEMENTATION DETAILS

We use a server with eight NVIDIA Tesla V100 GPUs for experiments. Since the pre-trained strategy may have the privileged knowledge of the testing distribution, all models are trained from scratch without loading any pre-trained weights to better evaluate the OOD generalization ability. The evaluation metrics—Average Precision (AP) and Average Recall (AR) are used, following the setting of MS COCO (Lin et al., 2014). For the Pascal VOC experiment, we follow the common setting of using VOC2007 trainval + VOC2012 trainval as our training set and VOC2007 test as our testing set to evaluate IID performance. For MS COCO, we randomly sample 10K images from MS COCO training set to optimize model parameters and evaluate on MS COCO val-set. Specifically, all training images from MS COCO are scaled to 640x640. And for BDD100K, which is comprised of 100K images for 1.8M objects of 10 categories, we use the image attribute labels to split OOD environments, where the train-test domains are non-overlapping. We choose 1K images from each training domain and 0.5K images from each test domain (more details can be found in Appendix A.2).

4.2 QUANTITATIVE RESULTS

IID dataset results. As illustrated in Table 1, we use Swin Transformer (Liu et al., 2021) as our baseline and compare our method to Cascade RCNN (Cai & Vasconcelos, 2018), RetinaNet (Lin et al., 2017), NAS-FPN (Ghiasi et al., 2019) and NAS-FCOS (Wang et al., 2020b) with ResNeXt-101 (Xie et al., 2017), ResNet-50 and ResNet-101 (He et al., 2016), respectively. Our method achieves the best performance of 46.6% AP with 153M parameters and 17.3% AP with 143M parameters on

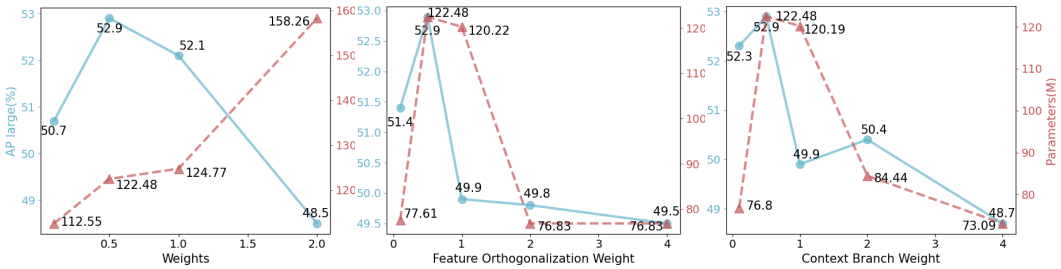


Figure 4: Performance of NAS-DO on BDD100K-weather with different context branch and feature orthogonalization weights, while super-net layer and step are set 4 and sparseness is 2. *left*: Context branch weight λ_{ctx} is equal to feature orthogonalization weight λ_p. *middle*: λ_{ctx} is fixed by 0.5. *right*: λ_p is fixed by 0.5.

Table 3: Ablation study for search space design on BDD100K-weather. Both context branch weight and feature orthogonalization weight are set to 0.5. Column P measures the searched backbone parameter size (M). For layer, step and sparseness are fixed by 4 and 1. For step, layer and sparseness are fixed by 4 and 2. For sparseness, layer and step are fixed by 4.

(a) Results of layer

Layer	P	AP_L	AR_L
4	139	46.0	56.3
5	110	42.9	51.7
6	**103**	**50.0**	**59.7**

(b) Results of step

Step	P	AP_L	AR_L
2	**57**	51.1	59.1
3	70	48.8	57.6
4	122	**52.9**	**59.8**

(c) Results of sparseness

Sparse	P	AP_L	AR_L
1	139	46.0	56.3
2	**122**	**52.9**	**59.8**

Pascal VOC and MS COCO, respectively.

OOD dataset results. As illustrated in Table 2, the proposed NAS-DO achieves the best performance of 52.9% AP_L on BDD100K-weather and outperforms baselines by nearly 20%. It also achieves the best performance of 39.8% AP_L on BDD100K-time-of-day and outperforms baselines 5.1% simultaneously. For the IID train-test split of BDD100K, we report the results in Appendix A.3 and NAS-DO consistently outperforms baseline methods with higher fps. Considering the comparative performance on IID and the outstanding generalization ability on OOD, we deduce that our searching strategy is able to find the optimal architecture with remarkable OOD generalization performance and reasonable model size since Pham et al. (2021) suggest there is a trade-off between model complexity and generalization ability and it needs larger models to achieve better OOD performance.

4.3 Ablation Study

Ablation studies are conducted to answer the following questions.

Q1: *The robustness of NAS-DO.*

A1: We study this by conducting corruption experiments where we train on clean data and test on corrupted data using image corruption tool-kits (Michaelis et al., 2019). The experimental results can be found in Appendix A.4, and it shows that NAS-DO achieves the best robustness among various corruption operations on our MS COCO benchmark.

Q2: *How much does the feature orthogonalization contribute to the improvement of generalization?*

A2: As shown in Figure 4, NAS-DO with 0.5 λ_{ctx} and λ_p, brings the generalization ability on BDD100K-weather up to 52.9% AP_L. Moreover, we set the λ_{ctx} and λ_p to 0 to learn the OOD generalization improvement contributed by feature orthogonalization and as shown in Table 2, NAS-DO outperforms baselines yet accuracy drops by 2.5% and 3.3% comparing to NAS-DO with 0.5 λ_{ctx} and λ_p on BDD100K-weather and time-of-day, respectively.

Q3: *What are the optimal weights of context branch and feature orthogonalization penalty?*

A3: We study the generalization impact of different λ_{ctx} and λ_p by setting the same value for both λ_{ctx} and λ_p, fixing one parameter and controlling the value of the other. Obviously, the results reported in Figure 4 demonstrate that when λ_{ctx} and λ_p are both set to 0.5, model achieves the optimal generalization ability.

Q4: *How much does the neural architecture search contribute to the improvement of generalization ability?*

A4: We study this by randomly sampling five architectures using different random seeds. As shown in Figure 5, model performance is limited under 32% AP_L without NAS, while optimal architecture converged by NAS brings the AP_L up to 39.8% with relatively lower parameter size.

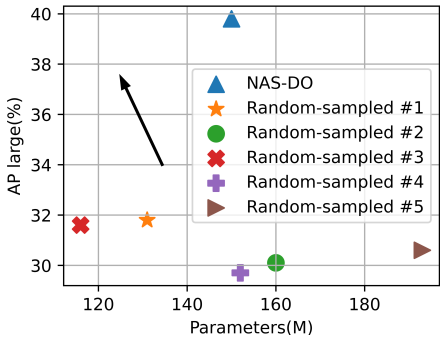


Figure 5: Ablation study of gradient-based searching strategy on BDD100K-time-of-day. #X indicates different random seeds.

Q5: *How do the width and the depth of super-net influence the searching process and performance?*

A5: First, as shown in Table 3(a), model performance achieves 50.0% AP_L with optimal 103M parameters in 6 layers and this turns out that deeper search space which means more sub-architectures is much more likely to discover not only the best function for fitting but also lesser parameters by doing variational optimization. Second, although models can reach the highest AP_L with step 4 (Table 3(b)), there exists a trade-off between complexity and performance. Lastly, Table 3(c) suggests that when increasing the sparseness which refers to more candidate sub-architectures, the model achieves the best performance when sparseness is set to 2.

Q6: *How do other OOD generalization algorithms work?*

A6: We use a Faster-RCNN with ResNet-50 backbone for detection and train the detection model with ERM and other OOD generalization algorithms, including IRM, vREx, GS, GroupDRO, and IGA on BDD100K-weather. The result is shown in Figure 2 and we can observe that the improvements of OOD generalization algorithms are only marginal over ERM. This further demonstrates the challenge of OOD generalization in object detection compared with the image classification task.

4.4 Disentanglement of Category-Related and Context-Related Features

We plot $W = \mathbf{1} - mean(\mathbb{1}(W_{cls} \neq \mathbf{0})) \cdot mean(\mathbb{1}(W_{ctx} \neq \mathbf{0}))$ in Figure 6 to analyze whether the feature orthogonalization successfully disentangle the category-related and context-related features, where $mean(\cdot)$ is element-wise mean function. Suppose we input a feature $F_i, i = 1, \ldots, n$ with n channels to the category branch and the context branch. If $W^i_{cls} = \mathbf{0}$ then $(W^i_{cls})^T \cdot F_i$ will be zero in category branch, while if $W^i_{cls} \neq \mathbf{0}$ then F_i can be used to predict the category labels. Same as W^i_{ctx}. $W_i = 1$ indicates that F_i is disentangled into the category branch or the context branch since, $W^i_{cls} \neq \mathbf{0}$ and $W^i_{ctx} = \mathbf{0}$, or, $W^i_{cls} = \mathbf{0}$ and $W^i_{ctx} \neq \mathbf{0}$ (we neglect both W^i_{cls} and $W^i_{ctx} = \mathbf{0}$ because this situation rarely happens). $W_i = 0$ indicates that F_i is used both to predict category label and context label ($W^i_{cls} \neq \mathbf{0}$ and $W^i_{ctx} \neq \mathbf{0}$) which means the feature has not been disentangled. It is obvious to see that at initialization, W_{cls} and W_{ctx} are hardly able to disentangle features since most channels of W are zero, after the training process finishes, W_{cls} and W_{ctx} can almost classify whether a feature is category-related or context-related as most channels of W are one compared to the initiation.

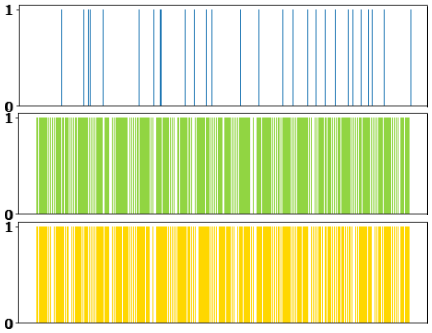


Figure 6: (**Top**) W_{cls} and W_{ctx} are initialized. (**Middle, Bottom**) W_{cls} and W_{ctx} are fixed after training under weather and time-of-day.

5 Conclusion and Discussion

In this paper, we propose NAS-DO, a novel feature-based neural architecture search framework for OOD object detection. We design a differentiable backbone super-net to search for the optimal detection backbone with the best OOD generalization ability guided by an orthogonal constraint on gradients of detector classifier heads to disentangle the category-related and context-related features. To the best of our knowledge, this is the first attempt to address NAS on OOD generalization object detection and simultaneously achieve the best performance. For future work, we will extend our method for real deployments.

REFERENCES

Kartik Ahuja, Karthikeyan Shanmugam, Kush Varshney, and Amit Dhurandhar. Invariant risk minimization games. In *International Conference on Machine Learning*, pp. 145–155. PMLR, 2020.

Martin Arjovsky, Léon Bottou, Ishaan Gulrajani, and David Lopez-Paz. Invariant risk minimization. *arXiv preprint arXiv:1907.02893*, 2019.

Hyojin Bahng, Sanghyuk Chun, Sangdoo Yun, Jaegul Choo, and Seong Joon Oh. Learning de-biased representations with biased representations. In *International Conference on Machine Learning*, pp. 528–539. PMLR, 2020.

Haoyue Bai, Rui Sun, Lanqing Hong, Fengwei Zhou, Nanyang Ye, Han-Jia Ye, S-H Gary Chan, and Zhenguo Li. Decaug: Out-of-distribution generalization via decomposed feature representation and semantic augmentation. *arXiv preprint arXiv:2012.09382*, 2020.

Haoyue Bai, Fengwei Zhou, Lanqing Hong, Nanyang Ye, S-H Gary Chan, and Zhenguo Li. Nas-ood: Neural architecture search for out-of-distribution generalization. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 8320–8329, 2021.

Han Cai, Ligeng Zhu, and Song Han. Proxylessnas: Direct neural architecture search on target task and hardware. *arXiv preprint arXiv:1812.00332*, 2018.

Zhaowei Cai and Nuno Vasconcelos. Cascade r-cnn: Delving into high quality object detection. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 6154–6162, 2018.

Fabio M Carlucci, Antonio D'Innocente, Silvia Bucci, Barbara Caputo, and Tatiana Tommasi. Domain generalization by solving jigsaw puzzles. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 2229–2238, 2019.

Kai Chen, Jiaqi Wang, Jiangmiao Pang, Yuhang Cao, Yu Xiong, Xiaoxiao Li, Shuyang Sun, Wansen Feng, Ziwei Liu, Jiarui Xu, Zheng Zhang, Dazhi Cheng, Chenchen Zhu, Tianheng Cheng, Qijie Zhao, Buyu Li, Xin Lu, Rui Zhu, Yue Wu, Jifeng Dai, Jingdong Wang, Jianping Shi, Wanli Ouyang, Chen Change Loy, and Dahua Lin. MMDetection: Open mmlab detection toolbox and benchmark. *arXiv preprint arXiv:1906.07155*, 2019a.

Yukang Chen, Tong Yang, Xiangyu Zhang, Gaofeng Meng, Xinyu Xiao, and Jian Sun. Detnas: Backbone search for object detection. *Advances in Neural Information Processing Systems*, 32: 6642–6652, 2019b.

Qi Dou, Daniel Coelho de Castro, Konstantinos Kamnitsas, and Ben Glocker. Domain generalization via model-agnostic learning of semantic features. *Advances in Neural Information Processing Systems*, 32:6450–6461, 2019.

M. Everingham, L. Van Gool, C. K. I. Williams, J. Winn, and A. Zisserman. The pascal visual object classes (voc) challenge. *International Journal of Computer Vision*, 88(2):303–338, June 2010.

Yaroslav Ganin, Evgeniya Ustinova, Hana Ajakan, Pascal Germain, Hugo Larochelle, François Laviolette, Mario March, and Victor Lempitsky. Domain-adversarial training of neural networks. *Journal of Machine Learning Research*, 17(59):1–35, 2016. URL `http://jmlr.org/papers/v17/15-239.html`.

Golnaz Ghiasi, Tsung-Yi Lin, and Quoc V Le. Nas-fpn: Learning scalable feature pyramid architecture for object detection. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 7036–7045, 2019.

Ishaan Gulrajani and David Lopez-Paz. In search of lost domain generalization. *CoRR*, abs/2007.01434, 2020. URL `https://arxiv.org/abs/2007.01434`.

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*, pp. 544–560. Springer, 2020.

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*, pp. 770–778, 2016.

Arthur Jacot, Clément Hongler, and Franck Gabriel. Neural tangent kernel: Convergence and generalization in neural networks. In *NeurIPS*, 2018.

Chenhan Jiang, Hang Xu, Wei Zhang, Xiaodan Liang, and Zhenguo Li. Sp-nas: Serial-to-parallel backbone search for object detection. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 11863–11872, 2020.

Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.

Masanori Koyama and Shoichiro Yamaguchi. Out-of-distribution generalization with maximal invariant predictor. 2021.

David Krueger, Ethan Caballero, Joern-Henrik Jacobsen, Amy Zhang, Jonathan Binas, Dinghuai Zhang, Remi Le Priol, and Aaron Courville. Out-of-distribution generalization via risk extrapolation (rex). In *International Conference on Machine Learning*, pp. 5815–5826. PMLR, 2021a.

David Krueger, Ethan Caballero, Joern-Henrik Jacobsen, Amy Zhang, Jonathan Binas, Dinghuai Zhang, Remi Le Priol, and Aaron Courville. Out-of-distribution generalization via risk extrapolation (rex). In *International Conference on Machine Learning*, pp. 5815–5826. PMLR, 2021b.

Clifford Law. The dangers of driverless cars. `https://www.natlawreview.com/article/dangers-driverless-cars`, 2021. May 5, 2021.

Da Li, Yongxin Yang, Yi-Zhe Song, and Timothy M Hospedales. Deeper, broader and artier domain generalization. In *Proceedings of the IEEE international conference on computer vision*, pp. 5542–5550, 2017.

Pan Li, Da Li, Wei Li, Shaogang Gong, Yanwei Fu, and Timothy M Hospedales. A simple feature augmentation for domain generalization. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 8886–8895, 2021.

Tingting Liang, Yongtao Wang, Zhi Tang, Guosheng Hu, and Haibin Ling. Opanas: One-shot path aggregation network architecture search for object detection. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 10195–10203, 2021.

Tsung-Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C Lawrence Zitnick. Microsoft coco: Common objects in context. In *European conference on computer vision*, pp. 740–755. Springer, 2014.

Tsung-Yi Lin, Priya Goyal, Ross Girshick, Kaiming He, and Piotr Dollár. Focal loss for dense object detection. In *Proceedings of the IEEE international conference on computer vision*, pp. 2980–2988, 2017.

Hanxiao Liu, Karen Simonyan, and Yiming Yang. Darts: Differentiable architecture search. *arXiv preprint arXiv:1806.09055*, 2018a.

Yen-Cheng Liu, Yu-Ying Yeh, Tzu-Chien Fu, Sheng-De Wang, Wei-Chen Chiu, and Yu-Chiang Frank Wang. Detach and adapt: Learning cross-domain disentangled deep representation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 8867–8876, 2018b.

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. *arXiv preprint arXiv:2103.14030*, 2021.

Gábor Melis, Chris Dyer, and Phil Blunsom. On the state of the art of evaluation in neural language models. *arXiv preprint arXiv:1707.05589*, 2017.

Claudio Michaelis, Benjamin Mitzkus, Robert Geirhos, Evgenia Rusak, Oliver Bringmann, Alexander S. Ecker, Matthias Bethge, and Wieland Brendel. Benchmarking robustness in object detection: Autonomous driving when winter is coming. *arXiv preprint arXiv:1907.07484*, 2019.

Xingchao Peng, Zijun Huang, Ximeng Sun, and Kate Saenko. Domain agnostic learning with disentangled representations. In *International Conference on Machine Learning*, pp. 5102–5112. PMLR, 2019.

Jonas Peters, Dominik Janzing, and Bernhard Schölkopf. *Elements of Causal Inference: Foundations and Learning Algorithms*. Adaptive Computation and Machine Learning. MIT Press, 2017. ISBN 978-0-262-03731-0. URL https://mitpress.mit.edu/books/elements-causal-inference.

Mohammad Pezeshki, Sékou-Oumar Kaba, Yoshua Bengio, Aaron Courville, Doina Precup, and Guillaume Lajoie. Gradient starvation: A learning proclivity in neural networks. *arXiv: Learning*, 2020.

Alan Pham, Eunice Chan, Vikranth Srivatsa, Dhruba Ghosh, Yaoqing Yang, Yaodong Yu, Ruiqi Zhong, Joseph E Gonzalez, and Jacob Steinhardt. The effect of model size on worst-group generalization. *arXiv preprint arXiv:2112.04094*, 2021.

Shaoqing Ren, Kaiming He, Ross Girshick, and Jian Sun. Faster r-cnn: Towards real-time object detection with region proposal networks. *Advances in neural information processing systems*, 28, 2015.

Shiori Sagawa, Pang Wei Koh, Tatsunori Hashimoto, and Percy Liang. Distributionally robust neural networks for group shifts: On the importance of regularization for worst-case generalization. *arXiv: Learning*, 2019.

Vladimir Vapnik. *Statistical Learning Theory*. Wiley, 1998.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Ł ukasz Kaiser, and Illia Polosukhin. Attention is all you need. In I. Guyon, U. Von Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett (eds.), *Advances in Neural Information Processing Systems*, volume 30. Curran Associates, Inc., 2017. URL https://proceedings.neurips.cc/paper/2017/file/3f5ee243547dee91fbd053c1c4a845aa-Paper.pdf.

Jingdong Wang, Ke Sun, Tianheng Cheng, Borui Jiang, Chaorui Deng, Yang Zhao, Dong Liu, Yadong Mu, Mingkui Tan, Xinggang Wang, et al. Deep high-resolution representation learning for visual recognition. *IEEE transactions on pattern analysis and machine intelligence*, 2020a.

Ning Wang, Yang Gao, Hao Chen, Peng Wang, Zhi Tian, Chunhua Shen, and Yanning Zhang. Nas-fcos: Fast neural architecture search for object detection. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 11943–11951, 2020b.

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*, pp. 1492–1500, 2017.

Hang Xu, Lewei Yao, Wei Zhang, Xiaodan Liang, and Zhenguo Li. Auto-fpn: Automatic network architecture adaptation for object detection beyond classification. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 6649–6658, 2019.

Shen Yan, Huan Song, Nanxiang Li, Lincan Zou, and Liu Ren. Improve unsupervised domain adaptation with mixup training. *arXiv:2001.00677*, 2020.

Yibo Yang, Hongyang Li, Shan You, Fei Wang, Chen Qian, and Zhouchen Lin. Ista-nas: Efficient and consistent neural architecture search by sparse coding. *arXiv preprint arXiv:2010.06176*, 2020.

Nanyang Ye, Kaican Li, Lanqing Hong, Haoyue Bai, Yiting Chen, Fengwei Zhou, and Zhenguo Li. Ood-bench: Benchmarking and understanding out-of-distribution generalization datasets and algorithms. *CoRR*, abs/2106.03721, 2021. URL https://arxiv.org/abs/2106.03721.

Fisher Yu, Wenqi Xian, Yingying Chen, Fangchen Liu, Mike Liao, Vashisht Madhavan, and Trevor Darrell. Bdd100k: A diverse driving video database with scalable annotation tooling. *arXiv preprint arXiv:1805.04687*, 2(5):6, 2018.

A APPENDIX

A.1 PROOFS

A.1.1 PROOF OF THEOREM A.4

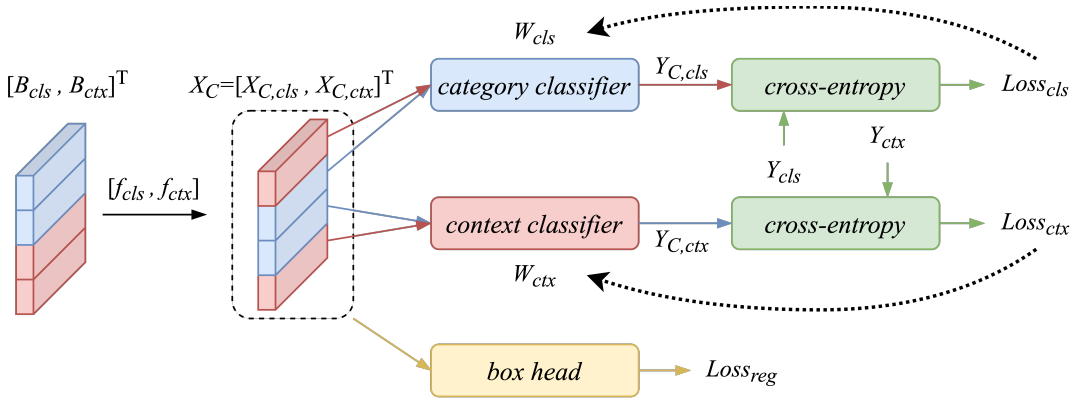


Figure 7: Illustration of the feature orthogonalization mechanism. Black dotted lines indicate the backward gradient. Blue blocks is the category features and Red blocks is the context features.

For completeness, the constraint, assumptions and main theorem are restated as followed. See Figure 7 for better understanding.

Assumption A.1. The category features B_{cls} and the context features B_{ctx} are independent $B_{cls} \perp\!\!\!\perp B_{ctx}$, and B_{cls} is independent to the context label Y_{ctx}, that is $B_{cls} \perp\!\!\!\perp Y_{ctx}$.

Assumption A.2. The input of the classifiers can be written as a concatenation (i.e. $X_C = [X_{C,cls}^T, X_{C,ctx}^T]^T$), where $X_{C,cls}$ is a function of the hidden category feature B_{cls}, (i.e. $\exists f_{cls} : \mathcal{R}^{B,cls} \to \mathcal{R}^{N_{C,cls}}$, $X_{C,cls} = f_{cls}(B_{cls})$), and $X_{C,ctx}$ is a function of the hidden context feature B_{ctx}, (i.e. $\exists f_{ctx} : \mathcal{R}^{B,ctx} \to \mathcal{R}^{N_{C,ctx}}$, $X_{C,ctx} = f_{ctx}(B_{ctx})$).

Constraint A.3. The weights of the category and context classifiers are orthogonal, that is

$$\mathbb{1}(W_{cls})^T \mathbb{1}(W_{ctx}) = \mathbf{0} \tag{12}$$

Theorem A.4. *(1) Theorem A.1 and Theorem A.2 hold; (2) the activation function is Lipschitz continuous; (3) the derivatives of the loss corresponding to the classifier outputs $Y_{C,cls}$ and $Y_{C,ctx}$, and the derivative of the activation function are stochastically bounded during the training; (4) the network widths goes to infinity; (5) the sample size goes to infinity. Then, Theorem A.3 is a sufficient condition for $Y_{C,cls} \perp\!\!\!\perp Y_{ctx}$.*

Proof. **Firstly**, according to NTK theorem Jacot et al. (2018), we use $W_{cls}(t)$ and $W_{ctx}(t)$ denote the W_{cls} and W_{ctx} at time t respectively for the purpose of representing the variation of the element in W_{cls} and W_{ctx} during the training process, then the dynamic of $W_{cls}(t)$ and $W_{ctx}(t)$ can be formulated as followed:

$$\partial_t W_{cls}(t) = -[\frac{\partial \mathcal{L}_{train}(t)}{\partial W_{cls}(t)}]^T \tag{13}$$

$$\partial_t W_{ctx}(t) = -[\frac{\partial \mathcal{L}_{train}(t)}{\partial W_{ctx}(t)}]^T \tag{14}$$

$$\mathcal{L}_{train} = \mathcal{L}_{cls} + \mathcal{L}_{reg} + \mathcal{L}_{ctx} + \mathcal{L}_{feat\_orth} \tag{15}$$

To simplify, we ignore the λ_{ctx} and λ_p in $\mathcal{L}_{train}$ and it is obvious that with the Theorem A.3, $\mathcal{L}_{feat\_orth}$ equals 0.

Secondly, we have the following deduction:

$$\frac{\partial \mathcal{L}_{reg}(t)}{\partial W_{cls}(t)} = \frac{\partial \mathcal{L}_{reg}(t)}{\partial W_{ctx}(t)} = 0 \tag{16}$$

$$\frac{\partial \mathcal{L}_{train}(t)}{\partial W_{cls}(t)} = X_C(t)^T X_C(t) W_{cls}(t) - X_C(t)^T Y_{cls} \tag{17}$$

$$\frac{\partial \mathcal{L}_{train}(t)}{\partial W_{ctx}(t)} = X_C(t)^T X_C(t) W_{ctx}(t) - X_C(t)^T Y_{ctx} \tag{18}$$

$$\tag{19}$$

and the weights matrices can be written as:

$$W_{cls}(t) = e^{-X_C^T X_C} W_{cls}(0) + \int_o^t e^{-X_C^T X_C \tau} d\tau \boldsymbol{X}_C(t)^T \boldsymbol{Y}_{cls} \tag{20}$$

$$W_{ctx}(t) = e^{-X_C^T X_C} W_{ctx}(0) + \int_o^t e^{-X_C^T X_C \tau} d\tau \boldsymbol{X}_C(t)^T \boldsymbol{Y}_{ctx} \tag{21}$$

$$\tag{22}$$

as $t \to \infty$, we have:

$$W_{cls}(\infty) = (\boldsymbol{X}_L^T \boldsymbol{X}_L)^{-1} \boldsymbol{X}_L^T \boldsymbol{Y}_{cls} \tag{23}$$

$$W_{ctx}(\infty) = (\boldsymbol{X}_L^T \boldsymbol{X}_L)^{-1} \boldsymbol{X}_L^T \boldsymbol{Y}_{ctx} \tag{24}$$

Thirdly, according to Theorem A.1 and Theorem A.2, we have $X_{C,cls} \perp\!\!\!\perp Y_{ctx}$, based on the Law of Large Number, $X_{C,cls} \perp\!\!\!\perp Y_{ctx}$ indicates $X_{C,cls}^T Y_{ctx} = \mathbf{0}$, thus as $t \to \infty$, we can write W_{ctx} as following:

$$W_{ctx} = \begin{bmatrix} \mathbf{0} \\ [f_{ctx}(B_{ctx})^T f_{ctx}(B_{ctx})]^{-1} f_{ctx}(B_{ctx})^T Y_{ctx} \end{bmatrix} = \begin{bmatrix} \mathbf{0} \\ [B_{ctx}^T B_{ctx}]^{-1} B_{ctx}^T Y_{ctx} \end{bmatrix} \tag{25}$$

After modifying Theorem A.3, W_{cls} can be written as:

$$W_{cls} = \begin{bmatrix} [B_{cls}^T B_{cls}]^{-1} B_{cls}^T Y_{cls} \\ \mathbf{0} \end{bmatrix} \tag{26}$$

Therefore, we have demonstrated that category prediction will not use the context information and Theorem A.3 is a sufficient condition for $Y_{C,cls} \perp\!\!\!\perp Y_{ctx}$. □

A.1.2 Proof of Theorem A.5

Theorem A.5. *Let $\mathcal{L}_{train}(\boldsymbol{\omega}, \boldsymbol{s})$ be continuous on $\mathbf{s}$ and $max\ \mathcal{L}_{train} \leq \infty$, then the sequence $\{z\}$ generated by Alg. 1 has limited points.*

Proof. For boundedness, it's obvious that $0 \leq \mathcal{L}_{train} \leq max\ \mathcal{L}_{train} \leq \infty$, thus $\mathcal{L}_{train}$ is bounded and $\mathcal{L}_{train}$ is closed set as well. For closedness, basically, $\mathcal{L}_{train}(\boldsymbol{\omega}, \mathbf{s})$ is continuous on $\mathbf{s}$, then the inverse image $\{\mathbf{s} | \mathcal{L}_{train}(\boldsymbol{\omega}, \mathbf{s})\}$ of a closed set $\mathcal{L}_{train}(\boldsymbol{\omega}, \mathbf{s})$ is closed. According to Heine-Borel Theorem, $\mathbf{s}$ is constrained within a compact sub-level set, then sequence $\{s\}$ has limited points, thus sequence $\{z\}$ generated by $\{s\}$ has limited points. □

A.2 Experimental Details of BDD100K

The original BDD100K contains 80000 labeled images (70000 for training and 10000 for validation) and each image has three attribute labels. We remove the images with the undefined attribute label and separate the rest into two OOD environments based on these attribute labels. See Table 5 for more details.

For optimization, We use SGD with 0.025 learning rate, 0.9 momentum and 0.0003 weight decay for optimizing network weights $\boldsymbol{\omega}$. We apply Adam (Kingma & Ba, 2014) with 0.0003 learning rate and 0.001 weight decay for optimizing architecture parameters $\mathbf{s}$. We use one sample per GPU, accounting for a batch size of eight. Object detectors are trained for 500 epochs on all experiments for convergence.

Table 4: More results on BDD100K datasets.

DETECTOR	OOD ENVIRONMENTS	PARAMS(M)	AP_L	AR_L	FPS
SWIN-B (LIU ET AL., 2021)	IID	145	36.5	45.0	11.6
NAS-DO		**124**	**42.3**	**50.3**	**13.0**
SWIN-B (LIU ET AL., 2021)	WEATHER	**145**	33.5	46.0	**11.6**
NAS-DO		166	**52.9**	**59.8**	9.3
SWIN-B (LIU ET AL., 2021)	TIME OF DAY	**145**	34.7	51.8	**11.6**
NAS-DO		150	**39.8**	**54.9**	9.2

Table 5: Details of BDD100K OOD environments training and testing set. **Sample quantity** indicates the number of the specific domain data sampled and **Quantity** indicates the total number of data in the original dataset. For training domains, we randomly sample at most 1500 pairs of data while at most 500 pairs for testing domains .

	OOD ENVIRONMENTS	TRAIN	TEST	SAMPLE QUANTITY	QUANTITY
CLEAR	WEATHER	√		1500	42690
OVERCAST		√		1500	10009
FOGGY			√	143	143
PARTLY CLOUDY			√	500	5619
RAINY			√	500	5808
SNOWY			√	500	6318
DAYTIME	TIME OF DAY	√		1500	41986
DAWN DUSK		√		1500	5805
NIGHT			√	500	31900

A.3 MORE EXPERIMENTAL RESULTS

Table 4 presents more experimental results on BDD100K (Yu et al., 2018) and the IID is the subset, with 10K for the training set and 5K for the testing set, of the original train-test split of BDD100K which is independent identically distributed.

A.4 CORRUPTION EXPERIMENTS

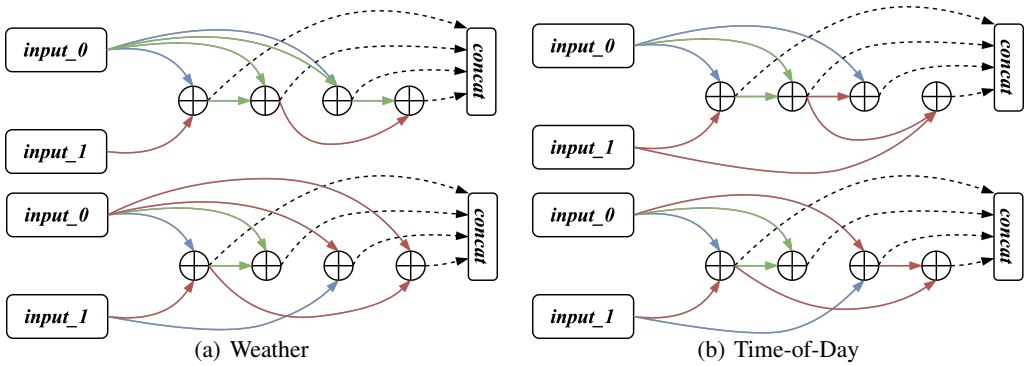
Table 6 presents the experimental results using image corruption tool-kits Michaelis et al. (2019) on MS COCO val-set corrupted by gaussian noise, shot noise, impulse noise, motion blur, zoom blur, brightness and contrast.

A.5 VISUALIZATION OF THE RESULTS

As illustrated in Figure 8, we present the details of the searched normal cell and reduction cell of NAS-DO. A cell contains two input nodes, four intermediate nodes and a concatenate layer. Each intermediate node has two edges pointed from the previous nodes and the chosen operation is presented on each edge in different colors. Moreover, the outputs of the four intermediate nodes are aggregated to the concatenate layer. Figure 9 presents some inference results on BDD100K-weather.

Table 6: Corruption experiment results on **MS COCO val-set** with various corruption methods. R-101 represents ResNet-101 backbone and X-101 represents ResNeXt-101.

MODEL	CORRUPTION	AP	AP_{50}	AP_{75}	AP_S	AP_M	AP_L
NAS-FPN @ X-101	GAUSSIAN NOISE	9.4	16.5	9.1	2.7	**9.5**	15.0
NAS-FPN @ R-101		8.6	15.5	8.5	**3.4**	8.9	13.8
SWIN-S		5.9	10.6	5.9	3.3	7.0	7.7
SWIN-B		5.9	10.3	5.9	3.2	6.8	8.5
NAS-DO (OURS)		**12.0**	**21.3**	**11.5**	0.4	6.2	**20.8**
NAS-FPN @ X-101	SHOT NOISE	9.7	16.9	9.6	2.4	**9.8**	15.6
NAS-FPN @ R-101		8.8	15.9	8.7	2.5	9.1	14.3
SWIN-S		6.3	11.3	6.2	**3.0**	7.2	8.5
SWIN-B		6.4	11.5	6.4	**3.0**	7.2	9.4
NAS-DO (OURS)		**12.6**	**22.6**	**11.9**	0.5	6.6	**22.2**
NAS-FPN @ X-101	IMPULSE NOISE	7.6	13.3	7.6	2.0	**7.8**	12.2
NAS-FPN @ R-101		6.9	12.5	6.7	2.6	7.0	11.4
SWIN-S		5.0	9.0	4.8	2.6	6.1	6.8
SWIN-B		5.1	8.9	5.2	**3.7**	6.3	7.2
NAS-DO (OURS)		**10.1**	**17.8**	**9.8**	0.3	4.9	**18.3**
NAS-FPN @ X-101	MOTION BLUR	11.9	20.9	11.9	3.0	**11.8**	19.2
NAS-FPN @ R-101		10.8	19.3	10.6	**3.3**	11.0	17.8
SWIN-S		7.7	14.6	7.3	2.8	8.1	11.8
SWIN-B		8.1	15.1	7.8	3.1	8.3	12.6
NAS-DO (OURS)		**13.8**	**24.4**	**13.4**	0.4	6.9	**24.3**
NAS-FPN @ X-101	ZOOM BLUR	5.7	11.9	5.0	**1.9**	**4.8**	10.2
NAS-FPN @ R-101		4.7	10.0	4.0	1.2	4.3	8.3
SWIN-S		2.3	5.0	1.8	1.0	2.3	4.0
SWIN-B		2.3	5.0	1.9	1.0	2.4	3.8
NAS-DO (OURS)		**5.8**	**12.5**	**4.3**	0.2	2.4	**10.4**
NAS-FPN @ X-101	BRIGHTNESS	12.7	21.9	12.7	3.6	**12.8**	19.7
NAS-FPN @ R-101		11.8	20.7	11.6	4.0	12.4	19.0
SWIN-S		11.1	20.1	10.9	5.5	11.6	15.0
SWIN-B		12.2	21.8	12.1	**6.2**	**12.8**	16.4
NAS-DO (OURS)		**15.1**	**26.0**	**14.9**	0.5	8.0	**26.2**
NAS-FPN @ X-101	CONTRAST	9.2	16.1	8.9	2.4	**9.3**	14.5
NAS-FPN @ R-101		8.2	14.5	8.0	2.7	8.7	13.6
SWIN-S		7.9	14.3	7.7	4.0	8.3	11.1
SWIN-B		8.2	14.7	8.1	**4.1**	8.8	11.3
NAS-DO (OURS)		**11.3**	**20.0**	**11.2**	0.5	6.5	**19.5**



(a) Weather (b) Time-of-Day

Figure 8: The searched normal cell (first row of each sub-figure) and reduce cell (second row of each sub-figure) of NAS-DO on BDD100K OOD environments. Red lines, green lines and blue lines represent $op_{0\sim2}$, respectively. Black dotted lines represent the output data flows. Better view in zoom-in mode.

(a) Swin Transformer (Baseline) (b) NAS-DO

Figure 9: Inference results of Swin Transformer and NAS-DO on BDD100K-weather environment with confidence threshold 0.7. Better view in zoom-in mode.