Object Detection with OOD Generalizable Neural Architecture Search

Fan Wu[1], **Kaican Li**[2], **Jinling Gao**[1], **Chensheng Peng**[1], **Lanqing Hong**[3], **Enze Xie**[3], **Zhenguo Li**[3], **Nanyang Ye**[1]*
[1] Shanghai Jiao Tong University [2] Hong Kong University of Science and Technology
[3] Huawei Noah's Ark Lab
wufan55@sjtu.edu.cn, mjust.lkc@gmail.com, gaojinling@sjtu.edu.cn,
pesiter-swift@sjtu.edu.cn, honglanqing@huawei.com, Johnny_ez@163.com,
li.zhenguo@huawei.com, ynylincoln@sjtu.edu.cn

Abstract

To improve the Out-of-Distribution (OOD) Generalization on Object Detection, we present a Neural Architecture Search (NAS) framework guided by feature orthogonalization. We believe that the failure to generalize on OOD data is due 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, such as "a car with four wheels", while the context-related features describe the non-causal information, such as "a car driving at night". However, due to the distinct data distribution between training and testing sets, the context-related features are often mistaken for causal information. To address this, we aim to automatically discover an optimal architecture that can 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 can achieve disentanglement and better performance on both IID and OOD.

1 Introduction

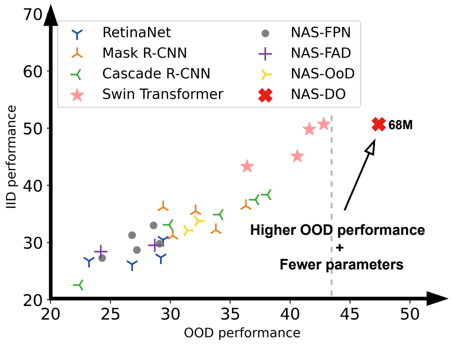

Figure 1: NAS-DO significantly outperforms baselines in terms of OOD performance with the fewest parameter size among SOTAs. Better view in zoom-in mode.

Object detection is a fundamental task in computer vision (Ren et al., 2015; Cai & Vasconcelos, 2018; Lin et al., 2017; Carion et al., 2020; Liu et al., 2021; Huang et al., 2019; Pang et al., 2019; Wu et al., 2019; Sun et al., 2020; Zhu et al., 2021; Tian et al., 2019; Wang et al., 2020; Ghiasi et al., 2019; Bochkovskiy et al., 2020; Ge et al., 2021; Zhang et al., 2020; Tan et al., 2020). 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).

To address the issue, we focus this paper on OOD generalization in object detection (OOD-OD). Currently, the literature on OOD-OD is still scarce as previous works on OOD generalization are

[*] Nanyang Ye is the corresponding author.

mostly devoted to image classification tasks. Although it might be possible to apply the same methodologies on object detectors, the hope of boosting their OOD performance is dim because the representative methods for classification only show very limited improvement beyond the classic empirical risk minimization (ERM) (Gulrajani & Lopez-Paz, 2020; Ye et al., 2021). Moreover, most current OOD algorithms (Arjovsky et al., 2019; Krueger et al., 2021) are built upon the domain-invariant principle, which assumes that invariant features can be learned to enable generalization across distributions. However, discovering such invariant features is challenging in object detection data, which are subject to large variations in viewpoints, lighting, and weather conditions. This can lead to severe over-fitting on the training dataset (see Appendix A.2 for further information).

Inspired by (Li et al., 2022) suggesting that the architectural design and the capacity of neural networks are crucial to OOD generalization, we explore the possibility of neural architecture search (NAS) for OOD-OD. There are many existing NAS methods for object detection (Jiang et al., 2020; Chen et al., 2019b; Wang et al., 2020; Ghiasi et al., 2019; Xu et al., 2019; Fang et al., 2020), but none of them have considered the OOD scenario. As we find out, those methods are not suited for OOD-OD as they aim to achieve maximal in-distribution performance, which tends to synergize with the easily over-fitting nature of NAS, sacrificing OOD performance.

In this work, we propose a specialized differentiable NAS framework, namely NAS-DO, for OOD-OD. The search process of NAS-DO is regularized by an OOD-aware objective called feature orthogonalization (FeatOrth) which favors architectures that are good at disentangling high-dimension object representations into category-related and context-related features. As previous study (Ye et al., 2021) suggests that category-related features are key to OOD generalization, FeatOrth, therefore, helps guide NAS-DO to discover architectures with great potential in OOD-OD. Figure 1 present the improvement of NAS-DO.

Our main contributions can be summarized as follows:

- To the best of our knowledge, our work is the first attempt to introduce NAS for OOD-OD, where the searching process is constrained by feature orthogonalization to obtain category-related information and context-related information.
- Extensive experiments demonstrate NAS-DO empirically outperforms previous SOTA baselines with up to 4.7% improvement on challenging OOD scenarios with fewer parameters.
- 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 METHODOLOGY

2.1 NAS FRAMEWORK

We base the differentiable searching framework on a two-stage detector, Cascade R-CNN (Cai & Vasconcelos, 2018), which consists of a backbone b, feature pyramid network (FPN), region proposal network (RPN) and prediction head h. The backbone b is replaced by our NAS super-net and is sequentially stacked by a set of searching cells $\{c_0, \dots, c_3\}$. Each cell is a normal cell or reduction cell and can be represented by a directed acyclic graph (DAG) consisting of n ordered nodes $X = \{x_1, x_2, \dots, x_n\}$ and edges between nodes $E = \{e^{(i,j)} | 1 \le i < j \le n\}$. The output of each edge is the concatenation of m candidate operations $O = \{o_0, o_1, \dots, o_{m-1}\}$. Binary variables $\alpha_k^{(i,j)} \in \{0, 1\}$ represent which operations will be active. Thus, we have the following formulations for each node:

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

where $\boldsymbol{\alpha}_j^T$ is converted into continuous $\mathbf{s}_j^T$ relaxation with a $softmax(\cdot)$ function. We apply a one-stage manner (Liu et al., 2018; Yang et al., 2020) with the architecture parameters constraint satisfied by formulating new architectures generating problem as a sparse coding problem to eliminate this performance gap:

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

where $A_j \in \mathbb{R}^{p_j \times (j-1)m}, p_j \leq (j-1)m$ denotes the measurement matrix and z_j is the sparse signal, which serves as a signal to terminate the searching process when **s** does not vary a lot. The outputs of the multi-level searching cells are passed to FPN for calculating the representations in different receptive scales.

2.2 Feature Orthogonalization

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

Assumption 2.1. 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 2.2. 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. $\exists f_{cls} : \mathcal{R}^{B,cls} \rightarrow \mathcal{R}^{N_{C,cls}}$, $X_{C,cls} = f_{cls}(B_{cls})$), and $X_{C,ctx}$ is a function of the hidden context feature B_{ctx}, (i.e. $\exists f_{ctx} : \mathcal{R}^{B,ctx} \rightarrow \mathcal{R}^{N_{C,ctx}}$, $X_{C,ctx} = f_{ctx}(B_{ctx})$).

Inspired by the above assumptions and to disentangle the extracted features, we design a two-branch detector head h_1 and h_2 , which consists 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]:

Constraint 2.1. The weights of the category and context classifiers are orthogonal, that is

$$\mathbb{1}(W_{cls})^T \mathbb{1}(W_{ctx}) = \mathbf{0} \tag{3}$$

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.* [2] Practically, we calculate the left-hand side of the Constraint 2.1 as the feature orthogonalization penalty $\mathcal{L}_{feat\_orth}$ during the training process.

2.3 Theoretical Analysis

Algorithm framework. Our searching process is outlined in Algorithm 1. Firstly, a super-net backbone and orthogonal heads are constructed for search. Then, we initialize the super-net parameters, including network weights $\boldsymbol{\omega}$ and architecture parameters **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. 2) and then derive the sparse sub-net $N_{S(z)}$. Lastly, network weights $\boldsymbol{\omega}$ and architecture parameters **s** are optimized by descending gradients using training loss. For the context branch, we adopt the same loss function as the category branch using image context labels:

$$\mathcal{L}_{ctx} = CE(Y_{ctx}(X), Y_{ctx}^*(X)) \tag{4}$$

where CE refers to the 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:

$$\mathcal{L}_{train} = \mathcal{L}_{RPN} + \mathcal{L}_{cls} + \mathcal{L}_{reg} + \lambda_{ctx} \cdot \mathcal{L}_{ctx} + \lambda_p \cdot \mathcal{L}_{feat\_orth} \tag{5}$$

where $\mathcal{L}_{RPN}$, $\mathcal{L}_{cls}$ and $\mathcal{L}_{reg}$ are consistent with (Cai & Vasconcelos, 2018), λ_{ctx} and λ_p are hyper-parameters to control the weights of $\mathcal{L}_{ctx}$ and $\mathcal{L}_{feat\_orth}$ in the whole training loss.

Disentanglement of feature orthogonalization. The efficiency of feature orthogonalization can be guaranteed by the following theorem:

[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 A.1.2)

[2] We apply the Straight Through Estimator (Courbariaux et al., 2016) to generate gradients for the indicator function, for more information please refer to Appendix A.5

Table 1: Comparison with SOTAs on the Weather-shift and Time-shift. AP_{iid} and AP_{ood} measure the IID and OOD performance. NAS-DO and NAS-OoD are both implemented on Cascade R-CNN detector (Cai & Vasconcelos, 2018). @X represents the inner dimension of NAS-OoD. Swin^{1,2} represent using Mask R-CNN and Cascade R-CNN structure introduced by the authors (Liu et al., 2021), while -T and -S represent the tiny and small version of Swin Transformer. The architectures of NAS-based methods are searched on Weather-OOD and Time-OOD for Weather-shift and Time-shift, respectively, and we report the average parameter size. #param. measures the parameter size in million. NAS-FAD and NAS-OoD are implemented by the authors and other baselines are implemented by mmdetection (Chen et al., 2019a).

method	backbone	#param.	Weather-shift		Time-shift		Avg_{iid}	Avg_{ood}
			AP_{iid}	AP_{ood}	AP_{iid}	AP_{ood}		
RetinaNet	ResNet-50	37M	24.6	24.8	29.0	21.6	26.8	23.2
RetinaNet	ResNet-101	55M	25.8	25.5	35.2	33.2	30.5	29.4
RetinaNet	ResNet-152	71M	24.3	24.0	30.4	34.3	27.4	29.2
RetinaNet	ResNeXt-50	57M	12.0	20.1	17.5	19.3	14.8	19.7
RetinaNet	ResNeXt-101	94M	26.8	27.8	25.6	25.8	26.2	26.8
Cascade R-CNN	ResNet-50	69M	30.6	29.6	35.6	30.1	33.1	29.9
Cascade R-CNN	ResNet-101	88M	31.4	30.9	38.3	37.2	34.9	34.1
Cascade R-CNN	ResNet-152	104M	34.6	32.7	40.3	41.5	37.5	37.1
Cascade R-CNN	ResNeXt-50	88M	20.1	20.3	25.0	24.4	22.6	22.4
Cascade R-CNN	ResNeXt-101	127M	34.9	35.2	41.9	41.0	38.4	38.1
SwinTransformer	Swin-T^1	48M	42.0	38.4	44.6	34.3	43.3	36.4
SwinTransformer	Swin-S^1	69M	42.8	42.0	47.3	39.1	45.1	40.6
SwinTransformer	Swin-T^2	86M	50.4	42.4	49.1	40.8	49.8	41.6
SwinTransformer	Swin-S^2	107M	**52.1**	43.7	49.3	41.8	**50.7**	42.8
NAS-FPN	ResNet-50	59M	31.7	30.2	34.3	26.9	33.0	28.6
NAS-FPN	ResNet-101	77M	28.1	25.0	29.2	29.3	28.7	27.2
NAS-FPN	ResNet-152	93M	30.1	23.5	32.4	30.1	31.3	26.8
NAS-FPN	ResNeXt-50	79M	25.7	24.9	33.8	33.2	29.8	29.1
NAS-FPN	ResNeXt-101	116M	23.0	22.1	31.6	26.5	27.3	24.3
NAS-FAD	ResNet-50	34M	16.2	13.7	17.4	17.3	16.8	15.5
NAS-FAD	ResNet-101	53M	26.5	25.1	30.2	23.3	28.4	24.2
NAS-FAD	ResNet-152	68M	29.2	28.4	29.8	29.0	29.5	28.7
NAS-FAD	ResNeXt-50	56M	19.7	12.1	18.4	12.0	19.1	12.1
NAS-FAD	ResNeXt-101	94M	11.2	10.7	15.0	10.1	13.1	10.4
NAS-OoD	NAS-OoD@d-36	47M	36.5	34.9	27.6	27.9	32.1	31.4
NAS-OoD	NAS-OoD@d-256	75M	37.8	36.1	29.8	28.4	33.8	32.3
NAS-DO	-	68M	51.6	**51.3**	**49.7**	**43.4**	**50.7**	**47.4**

Theorem 2.1. *(1) Assumption 2.1 and Assumption 2.2 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, Constraint 2.1 is a sufficient condition for $Y_{C,cls} \perp\!\!\!\perp Y_{ctx}$.*

We prove Theorem 2.1 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.3.1.

Convergence of the framework. The convergence of our proposed neural architecture search framework can be guaranteed by the following theorem:

Theorem 2.2. *Let $\mathcal{L}_{train}(\boldsymbol{\omega}, \boldsymbol{s})$ be continuous on* **s** *and $max\ \mathcal{L}_{train} \leq \infty$, then the sequence $\{z\}$ generated by Alg. 1 has limited points.*

The proof can be found in Appendix A.3.2.

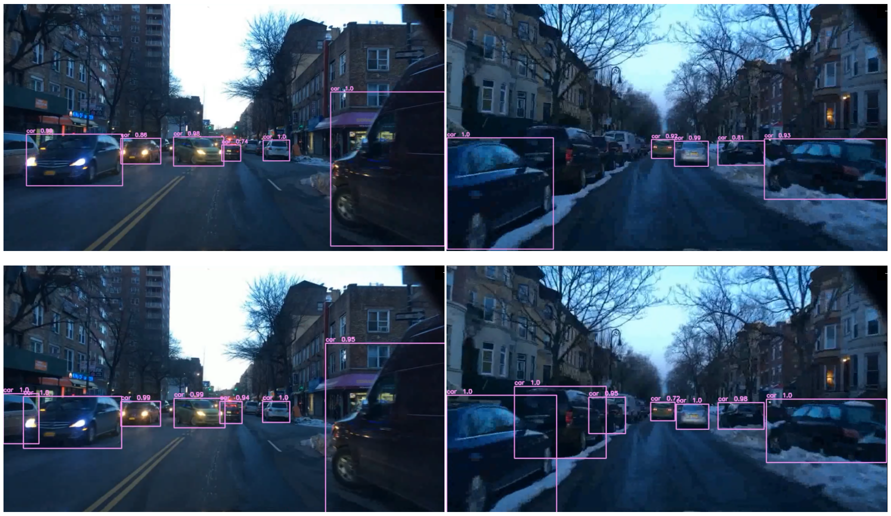

Figure 2: Inference results of Swin Transformer (Top) and NAS-DO (Bottom) on Weather-OOD with confidence threshold 0.7. Better view in zoom-in mode.

3 Experiments

3.1 OOD-OD Dataset Construction

We choose the BDD100K (Yu et al., 2018) dataset, which comprises 100K images of 1.8M objects categorized into 10 groups, including pedestrians, riders, cars, trucks, buses, trains, motorcycles, bicycles traffic lights and traffic signs, to construct OOD-OD datasets. We make use of image attribute labels provided by the official dataset to create multiple domains, such as daytime, dusk, night, etc. The details of domains in these datasets can be found in Appendix A.6. These labels specify the weather and time the image was captured. Based on these, we construct two OOD-OD datasets (Weather-shift and Time-shift). We also construct the No-shift counterpart for each dataset to evaluate methods' performance on IID and check the performance degeneration from IID to OOD.

3.2 Experimental results

Table 1 shows the results on Weather-shift and Time-shift datasets. Despite having smaller sizes, NAS-DO outperforms the baselines by achieving 51.3% and 43.4% with 68M parameters in OOD conditions, while baseline methods, such as RetinaNet (Lin et al., 2017), Mask R-CNN (He et al., 2017), Cascade R-CNN (Cai & Vasconcelos, 2018) and Swin Transformer (Liu et al., 2021), are susceptible to the subtle disturbance in data distribution as they lean to over-fit on the training set. Besides, other NAS-based methods are not suited for OOD-OD as they aim at finding the architecture with maximal in-distribution performance leading to even worse OOD-OD performance. Specifically, NAS-OoD applies a NAS strategy assisted by a conditional generator to solve OOD, however, it is hard to train an efficient conditional generator to generate object detection images, which usually comprise multiple objects and much more complicated, informative context with high resolution. These results demonstrate the superior OOD generalization ability of our proposed method with the NAS strategy guided by the FeatOrth regularization to avoid over-fitting on OOD-OD. Note that we give extra advantages to all the baselines by initializing their parameters using the weights pre-trained on the ImageNet-1K dataset (Russakovsky et al., 2015), which may contain data in the testing set.

4 Conclusion

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. pp. 145–155. PMLR, 2020.

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

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 *ICCV*, pp. 8320–8329, 2021.

Alexey Bochkovskiy, Chien-Yao Wang, and Hong-Yuan Mark Liao. Yolov4: Optimal speed and accuracy of object detection. *arXiv:2004.10934*, 2020.

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

Qi Cai, Yingwei Pan, Chong-Wah Ngo, Xinmei Tian, Lingyu Duan, and Ting Yao. Exploring object relation in mean teacher for cross-domain detection. In *CVPR*, pp. 11457–11466, 2019.

Zhaowei Cai and Nuno Vasconcelos. Cascade r-cnn: Delving into high quality object detection. In *CVPR*, pp. 6154–6162, 2018.

Nicolas Carion, Francisco Massa, Gabriel Synnaeve, Nicolas Usunier, Alexander Kirillov, and Sergey Zagoruyko. End-to-end object detection with transformers. In *ECCV*, 2020.

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.

Yuhua Chen, Wen Li, Christos Sakaridis, Dengxin Dai, and Luc Van Gool. Domain adaptive faster r-cnn for object detection in the wild. In *CVPR*, pp. 3339–3348, 2018.

Yukang Chen, Tong Yang, Xiangyu Zhang, Gaofeng Meng, Xinyu Xiao, and Jian Sun. Detnas: Backbone search for object detection. In *NeurIPS*, pp. 6642–6652, 2019b.

Matthieu Courbariaux, Itay Hubara, Daniel Soudry, Ran El-Yaniv, and Yoshua Bengio. Binarized neural networks: Training deep neural networks with weights and activations constrained to+ 1 or-1. *arXiv preprint arXiv:1602.02830*, 2016.

Qi Dou, Daniel Coelho de Castro, Konstantinos Kamnitsas, and Ben Glocker. Domain generalization via model-agnostic learning of semantic features. In *NeurIPS*, pp. 6450–6461, 2019.

Jiemin Fang, Yuzhu Sun, Kangjian Peng, Qian Zhang, Yuan Li, Wenyu Liu, and Xinggang Wang. Fast neural network adaptation via parameter remapping and architecture search. *arXiv preprint arXiv:2001.02525*, 2020.

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`.

Zheng Ge, Songtao Liu, Feng Wang, Zeming Li, and Jian Sun. YOLOX: Exceeding yolo series in 2021. *arXiv:2107.08430*, 2021.

Golnaz Ghiasi, Tsung-Yi Lin, and Quoc V Le. Nas-fpn: Learning scalable feature pyramid architecture for object detection. In *CVPR*, 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 *ECCV*, pp. 544–560. Springer, 2020.

Kaiming He, Georgia Gkioxari, Piotr Dollár, and Ross Girshick. Mask r-cnn. In *ICCV*, pp. 2961–2969, 2017.

Zeyi Huang, Haohan Wang, Eric P Xing, and Dong Huang. Self-challenging improves cross-domain generalization. In *ECCV*, pp. 124–140. Springer, 2020.

Zhaojin Huang, Lichao Huang, Yongchao Gong, Chang Huang, and Xinggang Wang. Mask scoring r-cnn. In *CVPR*, 2019.

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 *CVPR*, pp. 11863–11872, 2020.

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

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). pp. 5815–5826. PMLR, 2021.

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

Bo Li, Yifei Shen, Jingkang Yang, Yezhen Wang, Jiawei Ren, Tong Che, Jun Zhang, and Ziwei Liu. Sparse mixture-of-experts are domain generalizable learners. *arXiv preprint arXiv:2206.04046*, 2022.

Tingting Liang, Yongtao Wang, Zhi Tang, Guosheng Hu, and Haibin Ling. Opanas: One-shot path aggregation network architecture search for object detection. In *CVPR*, pp. 10195–10203, 2021.

Tsung-Yi Lin, Priya Goyal, Ross Girshick, Kaiming He, and Piotr Dollár. Focal loss for dense object detection. In *ICCV*, pp. 2980–2988, 2017.

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

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.

Jiangmiao Pang, Kai Chen, Jianping Shi, Huajun Feng, Wanli Ouyang, and Dahua Lin. Libra r-cnn: Towards balanced learning for object detection. In *CVPR*, 2019.

Xingchao Peng, Zijun Huang, Ximeng Sun, and Kate Saenko. Domain agnostic learning with disentangled representations. 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.

Shaoqing Ren, Kaiming He, Ross Girshick, and Jian Sun. Faster r-cnn: Towards real-time object detection with region proposal networks. In *NeurIPS*, 2015.

Olga Russakovsky, Jia Deng, Hao Su, Jonathan Krause, Sanjeev Satheesh, Sean Ma, Zhiheng Huang, Andrej Karpathy, Aditya Khosla, Michael Bernstein, et al. Imagenet large scale visual recognition challenge. *IJCV*, 115:211–252, 2015.

Peize Sun, Rufeng Zhang, Yi Jiang, Tao Kong, Chenfeng Xu, Wei Zhan, Masayoshi Tomizuka, Lei Li, Zehuan Yuan, Changhu Wang, and Ping Luo. SparseR-CNN: End-to-end object detection with learnable proposals. *arXiv:2011.12450*, 2020.

Mingxing Tan, Ruoming Pang, and Quoc V Le. Efficientdet: Scalable and efficient object detection. In *CVPR*, pp. 10781–10790, 2020.

Zhi Tian, Chunhua Shen, Hao Chen, and Tong He. Fcos: Fully convolutional one-stage object detection. *arXiv:1904.01355*, 2019.

Antonio Torralba and Alexei A. Efros. Unbiased look at dataset bias. In *CVPR*, 2011.

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.), *NeurIPS*, volume 30. Curran Associates, Inc., 2017. URL `https://proceedings.NeurIPS.cc/paper/2017/file/3f5ee243547dee91fbd053c1c4a845aa-Paper.pdf`.

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 *CVPR*, pp. 11943–11951, 2020.

Aming Wu and Cheng Deng. Single-domain generalized object detection in urban scene via cyclic-disentangled self-distillation. In *CVPR*, pp. 847–856, 2022.

Yue Wu, Yinpeng Chen, Lu Yuan, Zicheng Liu, Lijuan Wang, Hongzhi Li, and Yun Fu. Rethinking classification and localization for object detection. *arXiv:1904.06493*, 2019.

Chang-Dong Xu, Xing-Ran Zhao, Xin Jin, and Xiu-Shen Wei. Exploring categorical regularization for domain adaptive object detection. In *CVPR*, pp. 11724–11733, 2020.

Hang Xu, Lewei Yao, Wei Zhang, Xiaodan Liang, and Zhenguo Li. Auto-fpn: Automatic network architecture adaptation for object detection beyond classification. In *ICCV*, pp. 6649–6658, 2019.

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.

Haoyang Zhang, Ying Wang, Feras Dayoub, and Niko Sünderhauf. Varifocalnet: An iou-aware dense object detector. *arXiv:2008.13367*, 2020.

Xingxuan Zhang, Zekai Xu, Renzhe Xu, Jiashuo Liu, Peng Cui, Weitao Wan, Chong Sun, and Chen Li. Towards domain generalization in object detection. *arXiv preprint arXiv:2203.14387*, 2022.

Yangtao Zheng, Di Huang, Songtao Liu, and Yunhong Wang. Cross-domain object detection through coarse-to-fine feature adaptation. In *CVPR*, pp. 13766–13775, 2020.

Xizhou Zhu, Weijie Su, Lewei Lu, Bin Li, Xiaogang Wang, and Jifeng Dai. Deformable detr: Deformable transformers for end-to-end object detection. In *ICLR*, 2021. URL `https://openreview.net/forum?id=gZ9hCDWe6ke`.

A Appendix

A.1 Related Work

A.1.1 Object Detection

Recent object detection methods (Cai & Vasconcelos, 2018; Lin et al., 2017; Carion et al., 2020; Liu et al., 2021; Zhu et al., 2021; Pang et al., 2019; Huang et al., 2019) are mainly developed on an inherent assumption, *i.e.* , the training data and the test data are IID (Independent and Identically Distributed). However, models trained on IID dataset are susceptible to a subtle disturbance in test data distribution (Torralba & Efros, 2011). Domain Adaption (DA) methods (Chen et al., 2018; Cai et al., 2019; Xu et al., 2020; Zheng et al., 2020) are proposed to tackle the distribution gap by fine-tuning with the unsupervised testing domain images. These DA methods may fail when facing unseen data distributions in real scenarios. While the setting of OOD(domain) generalization for object detection is largely under-explored. Region Aware Proposal reweighTing (RAPT) (Zhang et al., 2022) is used to eliminate dependence within RoI features for domain generalization. Cyclic-Disentangled Self-Distillation (Wu & Deng, 2022) aims at disentangling domain-invariant representations. However, these works are short of considering the effect of architecture on OOD setting which may lead to sub-optimal performance.

Compared with NAS works for the standard image classification tasks, the works of NAS for object detection (Chen et al., 2019b; Jiang et al., 2020; Ghiasi et al., 2019; Liang et al., 2021; Xu et al., 2019; Wang et al., 2020; Fang et al., 2020) are relatively rare due to their intricacy. 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). Zhong *et al.* applies a differentiable searching strategy to effectively explore the optimal configuration of receptive fields for one-stage detectors (Fang et al., 2020). Ghiasi *et al.* designed a search space of scalable architecture to generate multi-scale feature representations (Ghiasi et al., 2019). 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). *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.*

A.1.2 OOD Generalization

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, these methods have been proven to show limited improvement in complex classification datasets (Gulrajani & Lopez-Paz, 2020; Ye et al., 2021) compared to empirical risk minimization and it is not easy to directly apply them to deal with OOD distribution shifts on the object detection task, which usually requires handling much more complex data. NAS-OoD (Bai et al., 2021) developed a conditional generator for classification to generate domain information, however, it is hard to train an efficient conditional generator to generate object detection images, which usually comprise multiple objects and much more complicated, informative context with high resolution.

A.2 Analysis of OOD algorithms

Figure 3 displays examples of object detection data that exhibit variations in viewpoint and lighting. Invariant-based OOD algorithms assume that causal features are invariant and can be learned to

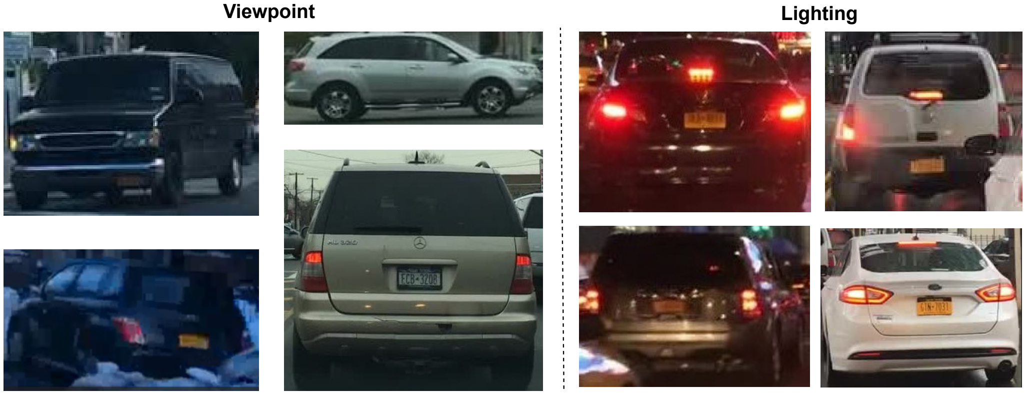

Figure 3: Variations in viewpoint and lighting.

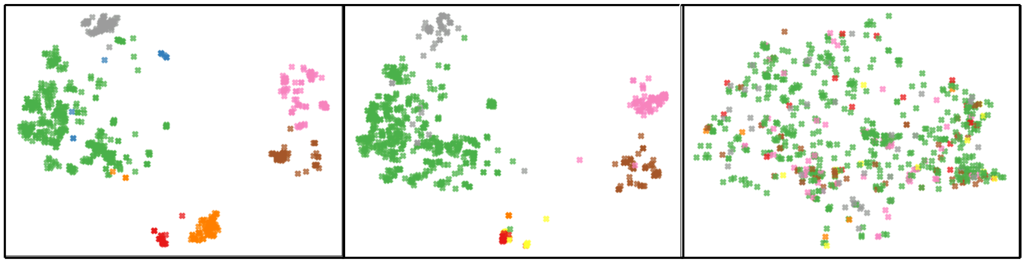

Figure 4: The t-SNE visualization of the features extracted by IRM (Arjovsky et al., 2019) on the Time-OOD dataset. Colors represent object categories. The two on the left are on the training domains and the right one is on the testing domain.

achieve generalization. However, identifying such invariant features is challenging given the significant variations that object detection data undergo, including changes in the viewpoint that can result in variant causal features. As shown in Figure 4, IRM is capable of learning similar feature patterns on training domains, however, this pattern fails to generalize to the testing domain, resulting in over-fitting. This over-fitting problem can be avoided by our proposed method. We leverage the high-capacity NAS regularized by FeatOrth, which favors architectures that are good at disentangling high-dimension object representations into category-related and context-related features, to avoid the easily over-fitting nature of NAS methods. Table 2 shows that our proposed NAS-DO surpasses IRM by 5.2% on the Time-OOD dataset.

A.3 Proofs

A.3.1 Proof of Theorem A.1

For completeness, the constraint, assumptions and main theorem are restated as followed. See Figure 5 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. $\exists f_{cls} : \mathcal{R}^{B,cls} \rightarrow \mathcal{R}^{N_{C,cls}}$, $X_{C,cls} = f_{cls}(B_{cls})$), and $X_{C,ctx}$ is a function of the hidden context feature B_{ctx}, (i.e. $\exists f_{ctx} : \mathcal{R}^{B,ctx} \rightarrow \mathcal{R}^{N_{C,ctx}}$, $X_{C,ctx} = f_{ctx}(B_{ctx})$).

Constraint A.1. The weights of the category and context classifiers are orthogonal, that is

$$\mathbb{1}(W_{cls})^T \mathbb{1}(W_{ctx}) = \mathbf{0} \tag{6}$$

Theorem A.1. *(1) Assumption A.1 and Assumption 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 $Y_{C,ctx}$,*

Table 2: Comparison with SOTA OOD algorithms. $\text{AP}^{\text{w}}_{\text{ood}}$ and $\text{AP}^{\text{t}}_{\text{ood}}$ measure the OOD performance on Weather-OOD and Time-OOD. Avg is the average performance on the two OOD scenarios. The results show that FeatOrth successfully makes the NAS process depart from sub-optimal OOD performance.

algorithm	#param.	$\text{AP}^{\text{w}}_{\text{ood}}(\%)$	$\text{AP}^{\text{t}}_{\text{ood}}(\%)$	Avg.
ERM (Vapnik, 1998)	61M / 63M	50.4	42.6	46.5
IRM (Arjovsky et al., 2019)	65M / 59M	49.4	38.2	43.8
VREx (Krueger et al., 2021)	74M / 59M	50.0	39.6	44.8
RSC (Huang et al., 2020)	69M / 65M	49.8	38.7	44.3
NAS-DO	68M / 67M	**51.3**	**43.4**	**47.4**

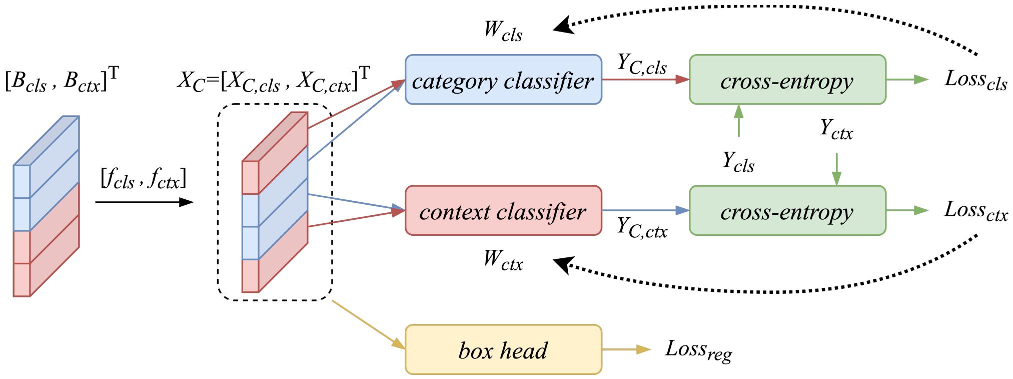

Figure 5: Illustration of the feature orthogonalization mechanism. Black dotted lines indicate the backward gradient. Blue blocks is the category features and Red blocks is the context features.

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, Constraint A.1 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:

$$\partial_t W_{cls}(t) = -[\frac{\partial \mathcal{L}_{train}(t)}{\partial W_{cls}(t)}]^T \tag{7}$$

$$\partial_t W_{ctx}(t) = -[\frac{\partial \mathcal{L}_{train}(t)}{\partial W_{ctx}(t)}]^T \tag{8}$$

$$\mathcal{L}_{train} = \mathcal{L}_{cls} + \mathcal{L}_{reg} + \mathcal{L}_{ctx} + \mathcal{L}_{feat\_orth} \tag{9}$$

To simplify, we ignore the λ_{ctx} and λ_p in $\mathcal{L}_{train}$ and it is obvious that with the Constraint A.1, $\mathcal{L}_{feat\_orth}$ equals 0.
Secondly, we have the following deduction:

$$\frac{\partial \mathcal{L}_{reg}(t)}{\partial W_{cls}(t)} = \frac{\partial \mathcal{L}_{reg}(t)}{\partial W_{ctx}(t)} = 0 \tag{10}$$

$$\frac{\partial \mathcal{L}_{train}(t)}{\partial W_{cls}(t)} = X_C(t)^T X_C(t) W_{cls}(t) - X_C(t)^T Y_{cls} \tag{11}$$

$$\frac{\partial \mathcal{L}_{train}(t)}{\partial W_{ctx}(t)} = X_C(t)^T X_C(t) W_{ctx}(t) - X_C(t)^T Y_{ctx} \tag{12}$$

$$\tag{13}$$

Algorithm 1: Object Detection with OOD Generalizable Neural Architecture Search

```
Input: training set $\mathcal{D}$, batch size $n$, learning rate $\beta$.
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. 2 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. 5.
    $\boldsymbol{\omega} \leftarrow \boldsymbol{\omega} - \beta \cdot \nabla \mathcal{L}_{train}(\mathcal{N}_{S(z)}(\boldsymbol{\omega}, \mathbf{s}))$.
    if $search\_flag$ then
      $\mathbf{s} \leftarrow \mathbf{s} - \beta \cdot \nabla \mathcal{L}_{val}(\mathcal{N}_{S(z)}(\boldsymbol{\omega}, \mathbf{s}))$.
    end if
  end for
  $z_{old} := z_{new}$.
end while
```

and the weights matrices can be written as:

$$W_{cls}(t) = e^{-X_C^T X_C} W_{cls}(0) + \int_o^t e^{-X_C^T X_C \tau} d\tau \boldsymbol{X}_C(t)^T \boldsymbol{Y}_{cls} \tag{14}$$

$$W_{ctx}(t) = e^{-X_C^T X_C} W_{ctx}(0) + \int_o^t e^{-X_C^T X_C \tau} d\tau \boldsymbol{X}_C(t)^T \boldsymbol{Y}_{ctx} \tag{15}$$

$$\tag{16}$$

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

$$W_{cls}(\infty) = (\boldsymbol{X}_L^T \boldsymbol{X}_L)^{-1} \boldsymbol{X}_L^T \boldsymbol{Y}_{cls} \tag{17}$$

$$W_{ctx}(\infty) = (\boldsymbol{X}_L^T \boldsymbol{X}_L)^{-1} \boldsymbol{X}_L^T \boldsymbol{Y}_{ctx} \tag{18}$$

Thirdly, according to Assumption A.1 and Assumption 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:

$$W_{ctx} = \begin{bmatrix} \mathbf{0} \\ [f_{ctx}(B_{ctx})^T f_{ctx}(B_{ctx})]^{-1} f_{ctx}(B_{ctx})^T Y_{ctx} \end{bmatrix} = \begin{bmatrix} \mathbf{0} \\ [B_{ctx}^T B_{ctx}]^{-1} B_{ctx}^T Y_{ctx} \end{bmatrix} \tag{19}$$

After modifying Constraint A.1, W_{cls} can be written as:

$$W_{cls} = \begin{bmatrix} [B_{cls}^T B_{cls}]^{-1} B_{cls}^T Y_{cls} \\ \mathbf{0} \end{bmatrix} \tag{20}$$

Therefore, we have demonstrated that category prediction will not use the context information and Constraint A.1 is a sufficient condition for $Y_{C,cls} \perp\!\!\!\perp Y_{ctx}$. □

A.3.2 PROOF OF THEOREM A.2

Theorem A.2. *Let $\mathcal{L}_{train}(\boldsymbol{\omega}, \boldsymbol{s})$ be continuous on* **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 **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, **s** is constrained within a compact sub-level set, then sequence $\{s\}$ has limited points, thus sequence $\{z\}$ generated by $\{s\}$ has limited points. □

Table 3: Details of the two constructed OOD-OD datasets. The IID and OOD conditions of each dataset are denoted as Weather(Time)-IID and Weather(Time)-OOD. "✓" represents the domain is chosen to construct training or testing domains.

dataset	domain	$\text{train}_{\text{iid}}$	$\text{train}_{\text{ood}}$	test
Weather-shift	clear		✓	
	overcast		✓	
	foggy	✓		✓
	cloudy	✓		✓
	rainy	✓		✓
	snowy	✓		✓
Time-shift	daytime		✓	
	dusk		✓	
	night	✓		✓

A.4 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.

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.

A.5 Differentiable Indicator Function

We implement the feature orthogonalization constraint based on Pytorch and inherit the torch.nn.Module. The gradient of the loss can be calculated during backward propagation. We apply the Straight Through Estimator (Courbariaux et al., 2016) to generate gradients for the indicator function. During the forward calculation, we use the indicator function to map the continuous input to $\{-1, 0, 1\}$. During the backward calculation, STE use the gradients of the continuous input to optimize parameters instead of gradients of the discrete outputs.

```
import torch
class LBSign(torch.autograd.Function):
    @staticmethod
    def forward(ctx, input):
        return torch.sign(input)

    @staticmethod
    def backward(ctx, grad_output):
        return grad_output.clamp_(-1, 1)
```

A.6 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. The details of the constructed OOD-OD datasets can be found in Table 3.

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.