GIFT: Representation Geometry Matters for Single-Domain Generalized Object Detection

Yin Zhang[1,2] **Yaoyue Zheng**[2,3]*
Yongqiang Zhang[4] **Bogdan Raducanu**[2] **Dan Liu**[1]

[1]School of Instrument Science and Engineering, Harbin Institute of Technology
[2]Computer Vision Center, Universitat Autònoma de Barcelona
[3]Institute of Artificial Intelligence and Robotics, Xi'an Jiaotong University
[4]College of Computer Science, Inner Mongolia University
{yin.zhang.hit, yaoyz105}@gmail.com
zhangyongqiang@imu.edu.cn, bogdan@cvc.uab.es, liudan@hit.edu.cn

Abstract

Single-Domain Generalized Object Detection (S-DGOD) aims to generalize from a single source domain to unseen domains under distribution shifts. Existing efforts mainly focus on simulating unseen domains or suppressing domain-specific factors. However, they largely overlook the intrinsic generalization capability embedded in Vision Foundation Models (VFMs). In this work, we show that unconstrained fine-tuning disrupts the geometric structure of pre-trained representations, which encodes domain-invariant semantics. Motivated by this, we propose **G**eometric-**I**nvariance **F**ine-**T**uning (**GIFT**), a geometric-aware fine-tuning method for VFMs. Specifically, we introduce a series of learnable Householder reflections to the pre-trained weights to encourage geometry-consistent adaptation under domain shift. In addition, we propose a lightweight channel-wise scaling to facilitate effective task-specific adaptation. Extensive experiments demonstrate that GIFT outperforms state-of-the-art (SOTA) approaches on two S-DGOD benchmarks. Notably, by introducing fewer than 1% additional trainable parameters into the frozen backbone, the proposed method achieves improvements of +7.9% and +2.9% in mPC over the previous SOTA on the Cityscapes-C and Diverse Weather Dataset, respectively. This finding highlights that preserving representation geometry is crucial for robust domain generalization, establishing GIFT as a principled and efficient solution to domain shift. The code is available at `https://github.com/zhangyin1996/GIFT`.

1 Introduction

Single-Domain Generalized Object Detection (S-DGOD) [51] aims to learn from a single source domain while generalizing to diverse unseen target domains. Traditional CNN-based methods primarily attempt to simulate domain diversity within a single source domain [55] or explicitly disentangle domain-invariant and domain-specific features [51]. Despite their effectiveness, these methods are generally built on CNN backbones such as ResNet [18] and pre-trained on medium-scale datasets (e.g., ImageNet [11]). Such constraints may limit representation capacity and hinder generalization to diverse unseen domains.

Recently, leveraging powerful Vision Foundation Models (VFMs), such as DINO [4, 37, 44], EVA [15, 14], and SAM [28, 40, 3], has become an effective strategy to overcome performance bottlenecks.

*Corresponding author.

40th Conference on Neural Information Processing Systems (NeurIPS 2026).

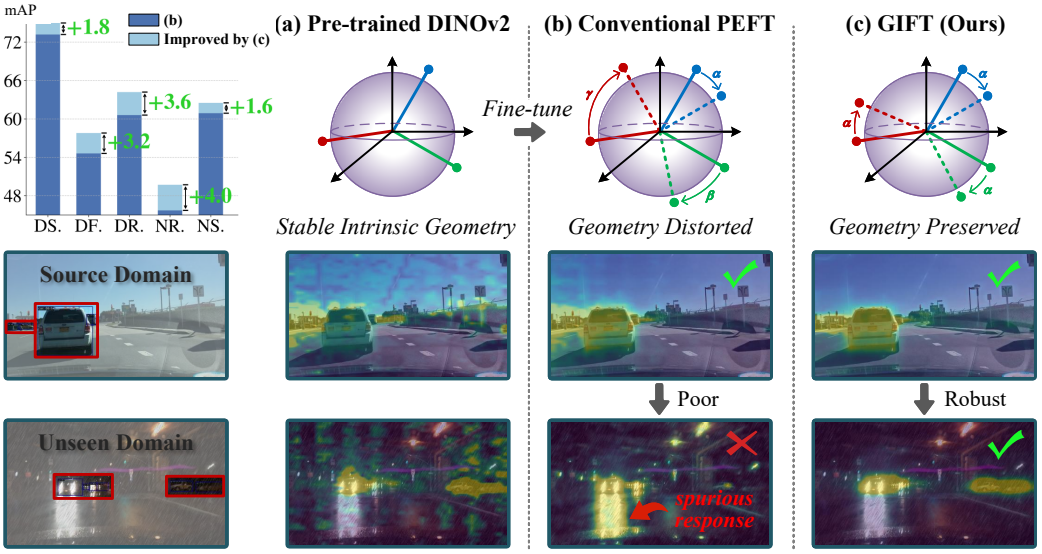

Figure 1: **(a)** The pre-trained VFM exhibits an intrinsically stable geometry, while **(b)** unconstrained fine-tuning [22] often disrupts this structure, degrading generalization. In contrast, **(c)** by preserving geometric structure during fine-tuning, GIFT effectively improves generalization across unseen domains (DF., DR., NR., and NS.) in the DWD dataset [51]. More results are provided in Section 4.4.

With large-scale pre-training and strong representation capacity, these models have significantly advanced various vision tasks [1, 59, 61]. A mainstream approach to exploiting these VFMs is Parameter-Efficient Fine-Tuning (PEFT) [17], which adapts the pre-trained model by updating only a small subset of parameters through lightweight modules such as prompts [25], adapters [6], or low-rank adaptations (LoRA) [20]. However, existing PEFT-based methods mainly focus on effectively adapting VFMs to downstream tasks. In the context of S-DGOD, the lack of multi-source data makes models prone to overfitting on source-specific statistics. Moreover, current PEFT-based S-DGOD methods largely follow conventional strategies from CNN-based methods, often overlooking the intrinsic generalization ability of VFMs, which is especially important under limited source diversity.

Motivation. We perform two experiments to show that unconstrained fine-tuning can disrupt the geometric structure of pre-trained representations, leading to degraded generalization on unseen target domains. By explicitly preserving this structure, our method improves cross-domain generalization.

Experiment 1: Attention Map Visualization. As shown in Fig. 1, we produce the mean CLS attention maps of the last layer [10, 24, 43] in DINOv2 for both the source and target domains of the DWD dataset [51], generated by **(a)** the pre-trained DINOv2, **(b)** an existing PEFT method [22], and **(c)** our fine-tuning method. Compared with the existing PEFT method, our method produces attention maps that more clearly highlight foreground objects in both the source and target domains. These observations suggest that preserving the geometric structure of pre-trained representations helps the model maintain consistent attention to foreground objects across different domains, which is beneficial for single-domain generalization.

Experiment 2: Geometric Statistical Analysis. We conduct a geometric statistical analysis to further support our motivation. First, for each input from Cityscapes-C [35], we feed the image into the DINOv2 [37] backbone and obtain the CLS token from the final transformer layer as the global representation. Then, inspired by PAID [48], we analyze the geometric properties of the representation space from three perspectives: Magnitude (M), Absolute Angle (A), and Structural Correlation (S). Specifically, given the feature matrix $F \in \mathbb{R}^{N \times D}$, where N represents the total number of image samples in the batch, and D denotes the feature dimension. The Magnitude (M) of each feature channel d is defined as:

$$M = \|F_{:,\,d}\|_2 \tag{1}$$

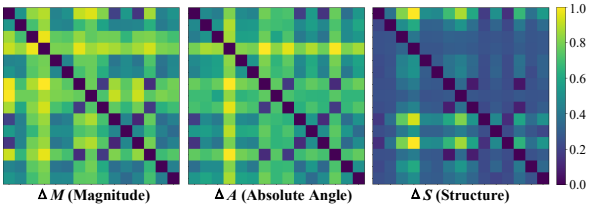

Figure 2: Geometric statistical analysis of DINOv2.

Table 1: The differences before and after fine-tuning for each PEFT method. '↓' indicates that lower values are better.

Methods	$D_M(\downarrow)$	$D_A(\downarrow)$	$D_S(\downarrow)$
LoRA [20]	**0.18**	0.085	0.11
Cauvis [30]	0.25	0.13	0.19
Ours	0.20	**0.051**	**0.081**

For absolute angle, we first normalize each channel vector: $\tilde{F}_{:,\,d} = F_{:,\,d}/\|F_{:,\,d}\|_2$, and we compute the global direction $\tau = \frac{1}{D}\sum_{d=1}^{D} \tilde{F}_{:,\,d}$, the Absolute Angle ($A$) of each feature channel d is:

$$A = \arccos(\tau^\top \tilde{F}_{:,\,d}) \tag{2}$$

where τ is L_2-normalized before computing the cosine angle. The Structural Correlation (S) between channels is defined as:

$$S = \tilde{F}^\top \tilde{F} \tag{3}$$

To quantify the representation differences between corruption domains, we compute pairwise distances of the above geometric descriptors. For two domains i and j, we define three metrics: Magnitude difference ΔM, Absolute Angle difference ΔA, and Structural difference ΔS. For fair visualization, all three geometric differences are independently normalized to [0,1] in Fig. 2.

In addition, we compare the geometric statistics of different PEFT methods before and after fine-tuning to provide stronger evidence. Specifically, using Eqs. (1)-(3), we compute the Magnitude (M), Absolute Angle (A), and Structural Correlation (S) of the representations before and after fine-tuning for each PEFT method, denoted as $(M_{\text{before}}, A_{\text{before}}, S_{\text{before}})$ and $(M_{\text{after}}, A_{\text{after}}, S_{\text{after}})$, respectively. Then, we quantify the relative changes in magnitude, absolute angle, and structural correlation before and after fine-tuning using normalized L_2 distances for M and A, and the normalized Frobenius distance for S, denoted as D_M, D_A, and D_S, respectively.

As shown in Fig. 2, the Magnitude (ΔM) and Absolute Angle difference (ΔA) exhibit substantial variations across corruption domains, whereas the Structural difference (ΔS) shows a more consistent pattern across different corruption pairs. Moreover, the results in Table 1 demonstrate that our method preserves representation geometry better than other PEFT methods after fine-tuning. This observation suggests that the structural relationships of DINOv2 representations are relatively stable under corruption shifts. However, the unconstrained fine-tuning of conventional adapter-based PEFT methods may disrupt this intrinsic geometric structure. Therefore, this motivates us to preserve the intrinsic geometric structure during fine-tuning.

Contribution. Building on these insights, we propose **G**eometric-**I**nvariance **F**ine-**T**uning (**GIFT**) for S-DGOD, which incorporates two key innovations. First, we preserve the geometric structure of pre-trained representations by introducing a series of Householder reflections [58], which ensure that the parameter space is transformed without altering its geometry. Second, we introduce a lightweight channel-wise scaling to the transformed weights to adapt task-specific features. Built on the geometric-preserving fine-tuning, this simple scaling yields noticeable gains in downstream object detection tasks, further highlighting the importance of maintaining the pre-trained geometric structure during fine-tuning. In summary, the main contributions of this paper are as follows:

- We are the first to investigate fine-tuning pre-trained VFMs for S-DGOD from a geometric perspective, revealing that unconstrained updates can distort representations and impair cross-domain generalization.
- We propose **GIFT**, a geometric-invariance fine-tuning method that preserves the intrinsic geometry of pre-trained representations, improving cross-domain generalization and enabling effective task-specific adaptation.
- Extensive experiments on two S-DGOD benchmarks demonstrate the effectiveness of GIFT, achieving +7.9% and +2.9% mPC improvements over the previous SOTA on Cityscapes-C and Diverse Weather Dataset, respectively.

2 Related Work

Single-Domain Generalized Object Detection (S-DGOD). Most S-DGOD methods improve generalization either by simulating domain diversity through augmentation/diversification [55, 9, 29], or by learning domain-invariant representations [32, 51, 34]. More recently, CLIP-based methods [39, 47, 54, 57] have shown promising results, but most methods still rely on full-parameter fine-tuning.

Leveraging the power of large pre-trained Vision Foundation Models (VFMs), PEFT [17, 53] has emerged as a promising direction for vision tasks [1, 46, 62]. However, S-DGOD under this paradigm remains largely underexplored, with only a few recent attempts [30, 59, 23]. Specifically, Cauvis [30] adopts DINOv2 [37] as a backbone and employs causal intervention to suppress spurious domain-specific correlations. Similarly, SoMA [59] applies Singular Value Decomposition to separate domain-specific dominant components from domain-invariant semantic components. However, these methods overlook the intrinsic generalization ability of VFMs. *In this work, we study PEFT-based S-DGOD from the perspective of the geometric structure of pre-trained representations. We reveal that maintaining a stable representation geometry is more critical for generalization than explicitly removing domain-specific features.*

Geometric Invariance in Representation Learning. Geometric invariance is crucial for representation learning. Early works such as SphereFace [33] and ArcFace [12] established this foundation by projecting features onto a hypersphere and optimizing angular margins, showing that angular distributions are more discriminative and robust. In domain-shift learning, AIDGN [26] and PAID [48] interpret angular structures as domain-invariant semantic cues. However, AIDGN necessitates inter-domain contrastive signals to identify invariant components, whereas PAID relies on access to target-domain data for adaptation. Recently, Householder Transformations [58, 13] have been proposed to preserve the geometric structure of the learnable feature space through orthogonal adaptations. However, they aim to improve adaptation efficiency rather than enhance cross-domain generalization. *Building on these insights, this work leverages geometric-preserving fine-tuning to tackle S-DGOD, enabling robust localization and recognition in unseen target domains.*

3 Proposed Method

3.1 Preliminaries

Task Formulation. Let $\mathcal{D}_s = \{(x_i, y_i)\}_{i=1}^{N_s}$ denote the source domain dataset, sampled from the distribution $(x, y) \sim P_s$. Our goal is to learn a detector that generalizes to any unseen target domain $\mathcal{D}_t \sim P_t$, where $P_s \neq P_t$.

In this work, we initialize the detector with a pre-trained model parameterized by $\boldsymbol{W}_0$ and fine-tune it on the source domain dataset $\mathcal{D}_s$. The training objective is formulated as:

$$\boldsymbol{W}_s = \arg\min_{\boldsymbol{W}} \mathbb{E}_{(x,y)\sim P_s}\big[\mathcal{L}(f_{\boldsymbol{W}}(x), y)\big] \tag{4}$$

where $\boldsymbol{W}_s$ is initialized from $\boldsymbol{W}_0$. In the S-DGOD setting, only $\mathcal{D}_s$ is accessible during training, while target domains remain unseen. Our objective is to learn source-optimal parameters $\boldsymbol{W}_s$ that generalize well to unseen target distributions P_t. Formally, we expect the target risk

$$\mathcal{R}_t(\boldsymbol{W}_s) = \mathbb{E}_{(x,y)\sim P_t}\big[\mathcal{L}(f_{\boldsymbol{W}_s}(x), y)\big] \tag{5}$$

to remain low for $P_t \neq P_s$.

Householder Reflection. To mitigate structural distortion, we adopt the Householder reflection [58] as a geometry-preserving adaptation mechanism. A Householder reflection refers to a linear transformation that performs a reflection with respect to a hyperplane defined by a unit vector $\boldsymbol{v} \in \mathbb{R}^d$, referred to as the Householder vector. The corresponding Householder matrix $\boldsymbol{H} \in \mathbb{R}^{d\times d}$ is constructed as follows:

$$\boldsymbol{H} = \boldsymbol{I} - 2\boldsymbol{v}\boldsymbol{v}^\top \quad \text{subject to } \|\boldsymbol{v}\|_2 = 1 \tag{6}$$

where $\boldsymbol{I}$ denotes the identity matrix and superscript $\top$ indicates the transpose of a matrix. A key geometric property of $\boldsymbol{H}$ is its **orthogonality** ($\boldsymbol{H}^\top\boldsymbol{H} = \boldsymbol{I}$), which ensures that the transformation is an isometry. For any two feature vectors $\boldsymbol{x}_1, \boldsymbol{x}_2$, the transformation preserves their inner product:

$$\langle \boldsymbol{H}\boldsymbol{x}_1, \boldsymbol{H}\boldsymbol{x}_2\rangle = (\boldsymbol{H}\boldsymbol{x}_1)^\top(\boldsymbol{H}\boldsymbol{x}_2) = \boldsymbol{x}_1^\top\boldsymbol{H}^\top\boldsymbol{H}\boldsymbol{x}_2 = \boldsymbol{x}_1^\top\boldsymbol{x}_2 \tag{7}$$

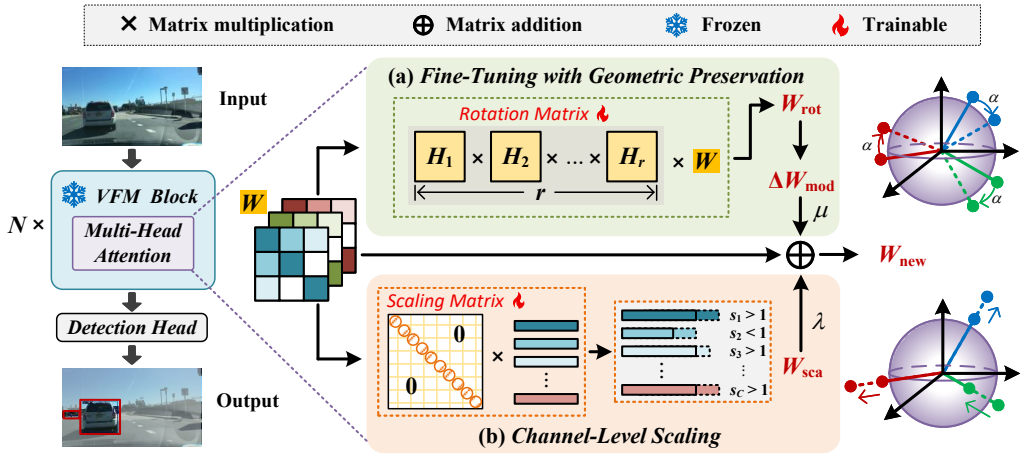

Figure 3: An overview of our proposed method. It is composed of two key components: **(a)** Fine-Tuning with Geometric Preservation (Section 3.2) and **(b)** Channel-Level Scaling (Section 3.3). Notably, we optimize only the multi-head attention parameters within each block. The updated weights, $\boldsymbol{W}_{\text{new}}$, are obtained by adding the scaled residual and channel-wise adjustment to the original weights $\boldsymbol{W}$.

This property forms the theoretical basis of our method, ensuring that geometric relationships among pre-trained feature representations remain stable during fine-tuning.

3.2 Fine-Tuning with Geometric Preservation

As shown in Fig. 3, for a frozen VFM with N Transformer blocks, we propose a geometric-preserving fine-tuning strategy that adapts the model while maintaining the intrinsic geometry of the pre-trained representation space. Instead of directly updating the backbone weights, we introduce a learnable rotation matrix $\boldsymbol{R}$, defined as the product of r Householder Reflection (HR) matrices, and integrate it into the frozen weights of the *Q*, *K*, *V*, and *Output* projection layers. The transformed weights $\boldsymbol{W}_{\text{rot}}$ can be represented as:

$$\boldsymbol{W}_{\text{rot}} = \boldsymbol{R}\boldsymbol{W} = \left(\prod_{i=1}^{r} \boldsymbol{H}_i\right)\boldsymbol{W} = \left(\prod_{i=1}^{r} (\boldsymbol{I} - 2\boldsymbol{v}_i\boldsymbol{v}_i^\top)\right)\boldsymbol{W} \tag{8}$$

where $\boldsymbol{H}_i = \boldsymbol{I} - 2\boldsymbol{v}_i\boldsymbol{v}_i^\top$ denotes the i-th HR matrix and $\boldsymbol{v}_i$ is a learnable unit vector. $\boldsymbol{W}$ denotes the original frozen projection weights. Since each HR matrix $\boldsymbol{H}_i$ is orthogonal, the product of orthogonal matrices remains orthogonal, thus the transformation $\boldsymbol{R}$ also satisfies $\boldsymbol{R}^\top\boldsymbol{R} = \boldsymbol{I}$. Consequently, for any pair of weight vectors $\boldsymbol{w}_1$ and $\boldsymbol{w}_2$, their inner product is preserved:

$$\langle \boldsymbol{R}\boldsymbol{w}_1, \boldsymbol{R}\boldsymbol{w}_2 \rangle = \langle \boldsymbol{w}_1, \boldsymbol{w}_2 \rangle \tag{9}$$

In particular, the ℓ_2 norm and pairwise distances remain invariant:

$$\|\boldsymbol{R}\boldsymbol{w}\|_2 = \|\boldsymbol{w}\|_2, \qquad \|\boldsymbol{R}\boldsymbol{w}_1 - \boldsymbol{R}\boldsymbol{w}_2\|_2 = \|\boldsymbol{w}_1 - \boldsymbol{w}_2\|_2 \tag{10}$$

These properties ensure that $\boldsymbol{R}$ rotates the parameter space without altering its geometry, thereby preserving the structural stability of the pre-trained model and mitigating distortions from direct weight updates.

In practice, directly applying $\boldsymbol{R}$ may induce large changes in the pre-trained weights. To ensure stable optimization, we employ a residual update mechanism. Specifically, we first compute the difference between the rotated and original weights, and then modulate this residual using a learnable gate vector $\boldsymbol{g} \in \mathbb{R}^{D_{out}}$, initialized to all ones so that all output channels contribute equally at the start of training:

$$\Delta\boldsymbol{W}_{\text{mod}} = \boldsymbol{g} \odot \Delta\boldsymbol{W} = \boldsymbol{g} \odot (\boldsymbol{W}_{\text{rot}} - \boldsymbol{W}) = \boldsymbol{g} \odot (\boldsymbol{R}\boldsymbol{W} - \boldsymbol{W}) \tag{11}$$

where $\odot$ denotes element-wise multiplication across columns of $\Delta\boldsymbol{W}$. The gate $\boldsymbol{g}$ allows selective adjustment of the residual along different directions, effectively controlling which components of the weight space are rotated.

3.3 Channel-Level Scaling for Stable Feature Selection

Preserving the pretrained semantic structure is necessary but not sufficient for single-domain generalization. To identify task-specific features that are consistent under source-domain variations, we introduce a learnable channel-wise scaling matrix $\boldsymbol{S}$, initialized to zeros:

$$\boldsymbol{W}_{\text{sca}} = \boldsymbol{S}\boldsymbol{W}, \quad \boldsymbol{S} = \text{diag}(s_1, \ldots, s_C) \tag{12}$$

where $\boldsymbol{W}_{\text{sca}}$ denotes the re-scaled weights, and each diagonal element of $\boldsymbol{S}$ modulates the importance of a single feature channel.

To measure feature stability, we adopt gradient consistency as a proxy. Let $f_i(x)$ denote the output of the i-th feature channel. The gradient of the detection loss $\mathcal{L}$ with respect to the scaling s_i is:

$$\mathcal{G}_i = \frac{\partial \mathcal{L}}{\partial f_i^{'}} f_i(x), \quad f_i^{'}(x) = s_i f_i(x) \tag{13}$$

The expected gradient over the source domain can be viewed as an indicator of how consistently channel i contributes across intra-domain variations. Channels that consistently contribute to loss reduction exhibit non-zero expected gradients, while spurious channels tend to produce gradients with high variance but near-zero mean. During stochastic optimization, gradients are accumulated over iterations:

$$s_i^T = s_i^0 - \eta \sum_{t=1}^{T} \mathcal{G}_i^t \tag{14}$$

where η denotes the learning rate. Under SGD optimization, channels with consistently aligned gradients tend to accumulate larger updates over training iterations, while unstable channels with highly varying gradients are less consistently reinforced.

The final updated weights $\boldsymbol{W}_{\text{new}}$ are obtained by adding the scaled residual to the original weights, together with a diagonal scaling branch to allow per-channel adjustment:

$$\boldsymbol{W}_{\text{new}} = \boldsymbol{W} + \mu\, \Delta \boldsymbol{W}_{\text{mod}} + \lambda\, \boldsymbol{W}_{\text{sca}} \tag{15}$$

where the balancing coefficients μ and λ are implemented as learnable scalar parameters.

Importantly, the orthogonal rotation branch primarily preserves relational geometric structure, such as pairwise angular relationships and relative orientations between pretrained weight vectors. In contrast, the channel-level scaling components intentionally modify feature magnitudes and channel importance to enable task-specific adaptation. Therefore, the proposed method should be viewed as preserving stable structural relations while allowing controlled feature reweighting, rather than enforcing strict isometry during adaptation.

4 Experiments

4.1 Benchmarks

Datasets and Evaluation Metrics. Following existing S-DGOD methods [63, 29, 30], we evaluate our method on two benchmarks: the Diverse Weather Dataset (DWD) [51] and Cityscapes-C [35]. **DWD** comprises five scenarios: Daytime-Sunny (DS.), Daytime-Foggy (DF.), Dusk-Rainy (DR.), Night-Rainy (NR.), and Night-Sunny (NS.). **Cityscapes-C** is a benchmark for evaluating detection robustness under common corruptions. It generates 15 corruption types with 5 severity levels.

For evaluation, we report mean Average Precision (mAP_{50}) to measure detection performance and mean Performance under Corruption (mPC) to evaluate robustness. For DWD [51], mPC is computed as the average mAP over four weather conditions, while for Cityscapes-C [35], it averages mAP_{50} across all corruption types and severity levels.

Implementation Details. We use DINOv2-Large [37] as the default VFM, while results with other VFMs are reported in Appendix A.3. For DWD, we employ Faster R-CNN [41] and DINO [60] (a DETR-based detector) as detection heads, respectively. For Cityscapes-C, Faster R-CNN is used as the detection head. All models are trained for 12 epochs using the AdamW optimizer with $\beta_1 = 0.9$ and $\beta_2 = 0.999$, a learning rate of 1×10^{-4}, and a batch size of 2. By default, the number of HR matrices is set to $r = 6$, and only the last $n = 16$ Transformer blocks are fine-tuned. All experiments are implemented in *MMDetection* [5] and conducted on a single RTX 3090 GPU.

Table 2: Comparison with SOTA methods on Cityscapes-C [35]. mPC denotes the average performance across 15 different corruption types. For fairness, we adopt Faster R-CNN [41] as the detection head. 'Baseline' denotes a fully frozen DINOv2 [37] backbone with only the Faster R-CNN detection head. '†' denotes unreported results re-implemented with the official code under the same settings.

		Noise			Blur				Weather				Digital				
Methods	Norm.	Gau.	Sho.	Imp.	Def.	Gla.	Mot.	Zoo.	Sno.	Fro.	Fog	Bri.	Con.	Ela.	Pix.	JPE.	**mPC**
Traditional S-DGOD																	
FRCNN [41]	42.2	0.5	1.1	1.1	17.2	16.5	18.3	2.1	2.2	12.3	29.8	32.0	24.1	40.1	18.7	15.1	15.4
AutoAug [63]	42.4	0.9	1.6	0.9	16.8	14.4	18.9	2.0	1.9	16.0	32.9	35.2	26.3	39.4	17.9	11.6	15.8
AugMix [19]	39.5	5.0	6.8	5.1	18.3	18.1	19.3	6.2	5.0	20.5	31.2	33.7	25.6	37.4	20.3	19.6	18.1
Stylized [16]	36.3	4.8	6.8	4.3	19.5	18.7	18.5	2.7	3.5	17.0	30.5	31.9	22.7	33.9	22.6	20.8	17.2
SupCL [27]	43.2	7.0	9.5	7.4	22.6	20.2	22.3	4.3	5.3	23.0	37.3	38.9	31.6	40.1	24.0	20.1	20.9
FSCE [45]	43.1	7.4	10.2	8.2	23.3	20.3	21.5	4.8	5.6	23.6	37.1	38.0	31.9	40.0	23.2	20.4	21.0
OA-DG [29]	43.4	8.2	10.6	8.4	24.6	20.5	22.3	4.8	6.1	25.0	38.4	39.7	32.8	40.2	23.8	22.0	21.8
Parameter-Efficient Fine-Tuning S-DGOD																	
Baseline	44.0	14.5	16.3	13.5	33.9	27.3	33.1	13.7	25.2	31.5	39.5	42.5	38.9	42.1	35.9	31.8	29.3
SoMA† [59]	53.5	16.4	19.2	15.3	39.4	31.4	37.7	15.0	28.2	36.7	47.2	51.5	45.9	50.9	42.3	36.3	34.2
Cauvis [30]	54.6	16.8	19.8	15.2	41.4	34.0	39.2	15.8	29.8	36.7	48.8	53.0	49.5	52.0	43.9	38.8	35.6
GIFT (Ours)	**62.7**	**23.2**	**25.1**	**20.6**	**49.8**	**43.2**	**48.0**	**19.2**	**39.8**	**44.2**	**58.2**	**61.4**	**58.0**	**61.0**	**53.3**	**48.1**	**43.5**

Table 3: Comparison with SOTA methods on DWD [51]. mPC denotes the average performance across 4 different adverse weather conditions.

Methods	DS.	DF.	DR.	NR.	NS.	**mPC**
Traditional S-DGOD						
IN [21]	43.9	28.5	22.8	12.6	29.6	23.4
SW [38]	50.6	30.8	26.3	13.7	33.4	26.1
ISW [7]	51.3	31.8	25.9	14.1	33.2	26.3
FRCNN [41]	56.2	33.5	28.0	14.2	35.8	27.9
CDSD [51]	56.1	33.5	28.2	16.6	36.6	28.7
OA-DG [29]	55.8	38.3	33.9	16.8	38.0	31.8
ClipGap [47]	52.5	38.5	32.3	18.7	36.9	31.6
G-NAS [52]	56.1	36.4	35.1	17.4	45.0	33.5
PDD [31]	53.6	39.1	33.7	19.2	38.5	32.6
UFR [34]	58.6	39.6	33.2	19.2	40.8	33.2
Parameter-Efficient Fine-Tuning S-DGOD						
Detection Head: Faster R-CNN [41]						
SoMA [59]	49.3	38.2	37.9	24.5	41.9	35.6
Cauvis [30]	72.2	52.3	61.3	44.1	58.3	54.1
GIFT (Ours)	**73.9**	**54.9**	**64.4**	**47.1**	**61.7**	**57.0**
Detection Head: DETR-based DINO [60]						
SoMA [59]	69.4	51.0	59.3	47.6	59.3	54.3
Cauvis [30]	73.7	56.5	64.6	47.6	61.2	57.5
GIFT (Ours)	**76.0**	**58.3**	**65.9**	**49.9**	**63.3**	**59.4**

Table 4: Ablation study of geometric preservation (GP) and channel-wise scaling (CS) on DWD, using DINO [60] as the detection head.

GP	**CS**	DF.	DR.	NR.	NS.	**mPC**
-	-	53.5	60.8	42.6	59.5	54.1
-	✓	57.1	62.9	46.8	60.7	56.9
✓	-	57.8	64.2	49.7	62.5	58.6
✓	✓	**58.3**	**65.9**	**49.9**	**63.3**	**59.4**

Table 5: Comparison of different PEFT methods on DINOv2 with DETR-based [60] detector.

Methods	DF.	DR.	NR.	NS.	**mPC**
Frozen	53.5	60.8	42.6	59.5	54.1
AdaFormer [6]	54.6	61.1	42.1	59.1	54.2
LoRA [20]	54.5	61.2	44.7	59.3	54.9
VPT [22]	54.6	60.6	45.7	60.9	55.5
Rein [50]	55.0	62.4	45.2	59.4	55.5
SPT [49]	55.7	62.6	46.6	60.6	56.4
CausalTune [61]	57.1	63.8	47.2	61.4	57.4
SoMA [59]	51.0	59.3	47.6	59.3	54.3
Cauvis [30]	56.5	64.6	47.6	61.2	57.5
GIFT (Ours)	**58.3**	**65.9**	**49.9**	**63.3**	**59.4**

4.2 Comparison with State-of-the-art S-DGOD Methods

Performance on Cityscapes-C. We evaluate the robustness of GIFT under diverse synthetic corruptions. For Cityscapes-C [35] dataset, the 15 corruption types are abbreviated as 'Gau.', 'Sho.', 'Imp.', 'Def.', 'Gla.', 'Mot.', 'Zoo.', 'Sno.', 'Fro.', 'Fog', 'Bri.', 'Con.', 'Ela.', 'Pix.', and 'JPE.'. 'Norm.' denotes the model's performance evaluated on the source domain, Cityscapes-Clean [8]. From Table 2, GIFT demonstrates clear superiority over existing methods in terms of overall performance (mPC). Specifically, GIFT achieves an average mPC of 43.5% across all 15 corruption types, outperforming the baseline (Frozen DINOv2 [37] with Faster R-CNN [41], +14.2%) and SOTA method Cauvis [30] (+7.9%).

Performance on DWD. The results in Table 3 demonstrate that our proposed method outperforms existing approaches across all adverse weather conditions and detector architectures. In the traditional S-DGOD setting, prior methods achieve limited robustness, with the best method (G-NAS) reaching an mPC of 33.5%. In contrast, parameter-efficient fine-tuning methods significantly improve performance, highlighting the effectiveness of adapting large pretrained models for S-DGOD. i) Under the Faster R-CNN detection head, our method GIFT achieves the best performance, reaching 57.0% in mPC and outperforming SoMA (35.6%) and Cauvis (54.1%). Notably, GIFT improves performance across all weather conditions (DF., DR., NR., NS.), indicating its strong generalization ability. ii)

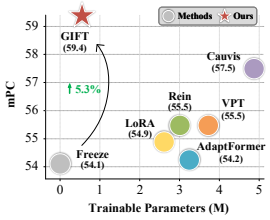

Figure 4: Trainable parameter efficiency of different methods based on DINOv2-Large [37].

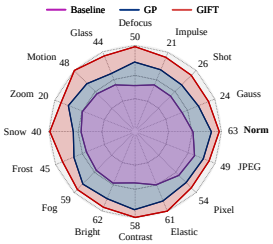

Figure 5: Ablation Studies on Cityscapes-C [35] dataset.

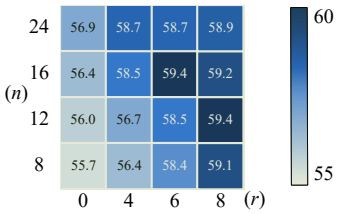

Figure 6: Sensitivity to the Rotation Rank r and Trainable Block Number n.

When using the DETR-based DINO detector, GIFT further improves performance, achieving an mPC of 59.4%, outperforming Cauvis (57.5%) by +1.9%.

4.3 Ablation Studies and Analysis

Effectiveness of Each Component. We present detailed ablation experiments on geometric preservation (**GP**) and channel-wise scaling (**CS**) across both benchmarks, shown in Table 4 for DWD and Fig. 5 for Cityscapes-C. The results show that both GP and GIFT (GP combined with CS) improve over the baseline across all target domains (4 for DWD and 15 for Cityscapes-C). Notably, larger gains on Cityscapes-C highlight their effectiveness under more challenging domain shifts. We also notice that CS further improves performance particularly when built upon a geometrically stable representation, underscoring the importance of preserving stable geometric representations for downstream task-specific adaptation.

Trainable Parameter Efficiency. To evaluate the training efficiency of GIFT, we compare the efficiency-effectiveness trade-offs of different methods built upon DINOv2-Large. As shown in Fig. 4, a fully frozen DINOv2-Large [37] backbone achieves 54.1% mPC with 0.0M trainable parameters, whereas GIFT improves mPC by 5.3% with only 0.52M parameters, demonstrating a superior performance-efficiency balance. Compared to other parameter-efficient fine-tuning strategies, including LoRA [20] (2.76M, 54.9% mPC), Rein [50] (2.99M, 55.5% mPC), AdaptFormer [6] (3.17M, 54.2% mPC), and VPT [22] (3.69M, 55.5% mPC), GIFT achieves higher performance while using 6-7$\times$ fewer parameters. Notably, it surpasses the SOTA method Cauvis [30] by +1.9% mPC while reducing trainable parameters by nearly 90%. These results highlight that GIFT not only enhances robustness but also maintains exceptional parameter efficiency.

Sensitivity to the Rotation Rank r and Trainable Block Number n. We investigate the impact of the rotation rank r and the number of trainable blocks n on DWD dataset.[1] Fig. 6 reports the mPC results across the four adverse weather conditions. We vary r from 0 to 8, where $r = 0$ indicates that no rotation matrix is applied. For n, we evaluate a wide range from 8 to 24. We observe that the performance generally improves as r increases. However, the gains gradually saturate as r approaches 8, suggesting that a moderately small r is sufficient to capture most of the geometric invariance. We also observe that the performance improves as n increases, reaching the best results when n is between 16 and 24. Based on these observations, we adopt $r = 6$ and $n = 16$ as the default settings in all main experiments, achieving a favorable balance between performance and computational cost.

Feature Correlation Analysis. Following OA-DG [29], we measure the feature correlation between the clean and corrupted domains on the Cityscapes [8] and Cityscapes-C [35] to analyze the stability of learned representations under domain corruption. Please refer to the Appendix B for more details. We visualize the correlation matrices as heatmaps using five corruption types in Cityscapes-C as examples: 'Gaussian noise', 'Impulse noise', 'Defocus blur', 'Motion blur' and 'Elastic transform'. As shown in Fig. 7, the correlation matrices exhibit a strong diagonal structure across all corruption types. This indicates that the class-wise representations learned by the model remain largely consistent under various corruption shifts, suggesting that our method could preserve geometric structure of pre-trained representations under domain perturbations.

[1] No target-domain data or labels were used. All configurations were predefined and trained on source-domain data.

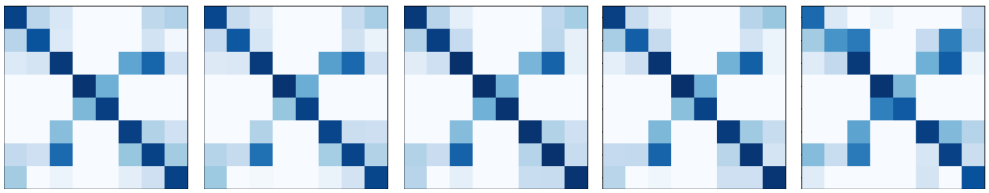

Figure 7: Feature correlation analysis on Cityscapes [8] (Clean domain) and Cityscapes-C [35] (Corrupted domain) datasets. Darker colors indicate higher cosine similarity between class features.

Table 6: Comparison of Training Time (s/iter) and Inference Throughput (images/s)

Methods	Training Time ↓	Inf. Throughput ↑	**mPC**
LoRA [20]	**0.72**	**6.31**	54.9
Cauvis [30]	1.93	2.78	<u>57.5</u>
GIFT (Ours)	<u>0.83</u>	<u>5.21</u>	**59.4**

Table 7: Comparison of different update schemes for $\boldsymbol{W}$ on Cityscapes-C.

Schemes	Res.	$\boldsymbol{g}$	Source	Target
$\boldsymbol{W} \rightarrow \boldsymbol{W}_{\text{rot}}$	-	-	55.8	34.2
$\boldsymbol{W} \rightarrow \Delta\boldsymbol{W}$	✓	-	<u>57.3</u>(+1.5↑)	<u>36.5</u>(+2.3↑)
$\boldsymbol{W} \rightarrow \Delta\boldsymbol{W}_{\text{mod}}$	✓	✓	**58.2**(+0.9↑)	**37.6**(+1.1↑)

Comparison of Training Time and Inference Throughput. Table 6 compares training efficiency and inference speed across different methods. LoRA achieves the fastest training time (0.72 s/iter) and highest inference throughput (6.31 images/s), but its performance is lower than the others. Cauvis improves accuracy (57.5% in mPC) but at the cost of significantly slower training and inference. In contrast, our method strikes a better balance, achieving competitive efficiency (0.83 s/iter, 5.21 images/s) while obtaining the best performance (59.4% in mPC).

Comparison of Different Update Schemes. In Eq. (11), we employ a residual update mechanism and utilize a learnable gate vector $\boldsymbol{g}$ to ensure stable optimization. We analyze the impact of different update mechanisms of the weight $\boldsymbol{W}$ on the Cityscapes-C [35]. As shown in Table 7, compared with directly introducing a learnable rotation matrix ($\boldsymbol{W} \rightarrow \boldsymbol{W}_{\text{rot}}$), applying only a residual update mechanism ($\boldsymbol{W} \rightarrow \Delta\boldsymbol{W}$) yields an improvement of +1.5% mAP and +2.3% mPC, respectively. Moreover, introducing the gate $\boldsymbol{g}$ could further improve the performance in the source (from 57.3% mAP to 58.2% mAP, +0.9%) and the target (from 36.5% mPC to 37.6% mPC, +1.1%) domain.

Comparison with Other PEFT Methods. Table 5 presents a comparison of our method against various PEFT strategies on the DINOv2-Large backbone. Compared with simple freezing (54.1%), all PEFT methods yield performance improvements, confirming the necessity of PEFT for S-DGOD. Importantly, our method consistently achieves the best performance across all conditions and obtains the highest overall mPC of 59.4%. For instance, under the most challenging 'Night-Rainy' (NR.) conditions, our method reaches 49.9% in mAP, showing a substantial improvement over LoRA [20] (44.7%, +5.2%). Furthermore, our method surpasses SOTA method Cauvis [30] by +1.9% in mPC, achieving robust improvements under all four conditions.

4.4 Visualization Results

Visualization of Attention Maps. To further verify the importance of geometric structure of pre-trained representations, we visualize the attention maps of the pre-trained DINOv2 [37], conventional PEFT methods (AdaptFormer [6] and VPT [22]), and our method on the DWD [51]. From Fig. 8 we can see that the attention maps of the pre-trained DINOv2 contain many artifacts [10]. Conventional PEFT methods AdaptFormer [6] and VPT [22] fail to properly focus on foreground objects, as unconstrained fine-tuning may disrupt the geometric structure of the pre-trained representations. Notably, the attention maps of ours highlight the target regions more clearly, demonstrating that preserving the geometric structure of pre-trained representations is important for domain generalization.

Visualization of Detection Results. To show the effectiveness of our method, we provide some visualization results on DWD [51]. As shown in Fig. 9, existing methods exhibit various detection errors. For example, in Fig. 9 (d), ClipGap [47] fails to detect the truck on the right. Some methods produce false detections of non-existent objects. For example, the baseline detects a non-existent

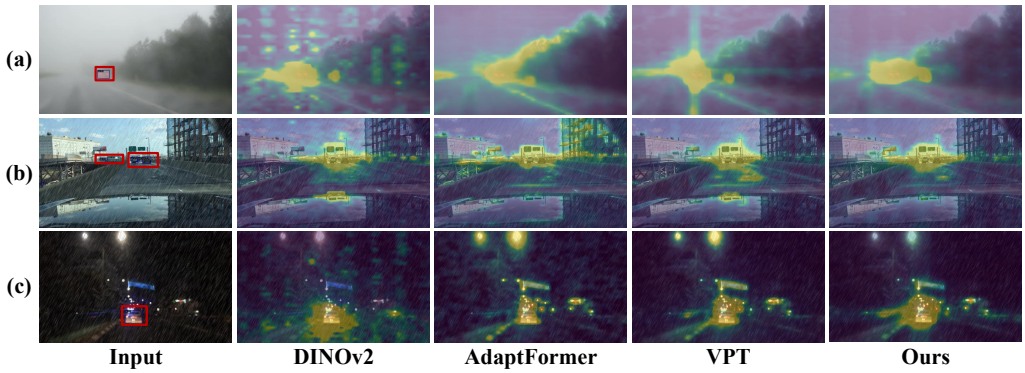

Figure 8: Visualization of attention maps for the DINOv2 [37], conventional PEFT methods AdaptFormer [6], VPT [22], and our geometric-preserving (GP) fine-tuning strategy on the DWD [51].

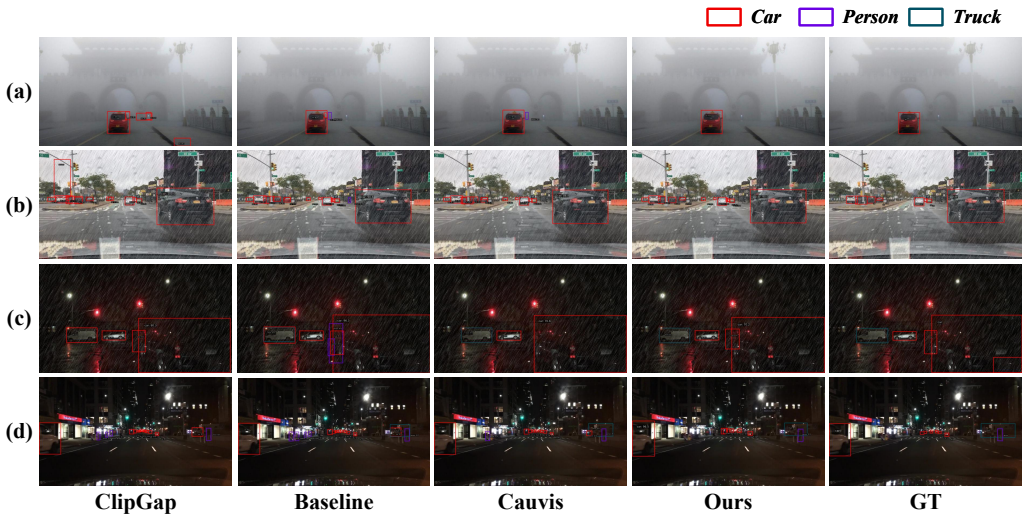

Figure 9: Visualization of detection results on the DWD [51]. (a) to (d) correspond to the four weather conditions of the DWD target domain: DF., DR., NR., and NS. GIFT is compared with the traditional S-DGOD method ClipGap [47], Baseline (Frozen DINOv2 [37] with Faster R-CNN [41]) and the SOTA method Cauvis [30]. Best seen on computer, in color and zoomed in.

person under daytime-foggy, night-rainy, and night-sunny conditions (see Fig. 9(a), (c), (d)). In addition, the SOTA method Cauvis [30] suffers from the same issue (Fig. 9). Furthermore, other methods misclassify objects, such as in Fig. 9(c), where the baseline mis-classifies a truck as a car. In contrast, our method achieves reliable detection performance across all four weather conditions, even in the most challenging night-rainy (NR.) scenario. More results can be found in the Appendix D.

5 Conclusion

In this work, we revisit Single-Domain Generalized Object Detection (S-DGOD) from the perspective of representation geometry in Vision Foundation Models. We show that unconstrained fine-tuning disrupts pre-trained geometric structures, undermining domain-invariant generalization. To address this, we propose Geometric-Invariance Fine-Tuning (GIFT), a simple yet principled method that preserves representation geometry while enabling effective task adaptation. Experiments on two S-DGOD benchmarks demonstrate that GIFT consistently surpasses state-of-the-art methods, achieving significant performance gains with fewer than 1% additional trainable parameters. These results highlight the importance of preserving pre-trained representation geometry for robust domain generalization and suggest a promising direction for extending GIFT to broader vision tasks.

6 Acknowledgments

This work was conducted during the author's research visit to the Computer Vision Center (CVC). The author sincerely thanks Yaoyue Zheng, Bogdan Raducanu, Joost van de Weijer and many other colleagues in the Learning and Machine Perception (LAMP) team at CVC for their valuable suggestions and discussions.

This work was supported by Grants PID2025-175937NB-I00 and AIA2025-163919-C52 funded by MICIU/AEI/10.13039/501100011033, and by the Generalitat de Catalunya CERCA Program. Yongqiang Zhang acknowledges the National Natural Science Foundation of China (No. 62666051), the Inner Mongolia Natural Science Foundation for Distinguished Young Scholars (No. 2025JQ009), and the Inner Mongolia Talent Development Project for Outstanding Young Talents. Yaoyue Zheng acknowledges the China Scholarship Council (CSC) No.202406280387.

References

[1] Nermeen Abou Baker, David Rohrschneider, and Uwe Handmann. Parameter-efficient fine-tuning of large pretrained models for instance segmentation tasks. *Machine Learning and Knowledge Extraction*, 6(4):2783–2807, 2024.

[2] Qi Bi, Jingjun Yi, Hao Zheng, Haolan Zhan, Yawen Huang, Wei Ji, Yuexiang Li, and Yefeng Zheng. Learning frequency-adapted vision foundation model for domain generalized semantic segmentation. *Advances in Neural Information Processing Systems*, 37:94047–94072, 2024.

[3] Nicolas Carion, Laura Gustafson, Yuan-Ting Hu, Shoubhik Debnath, Ronghang Hu, Didac Suris, Chaitanya Ryali, Kalyan Vasudev Alwala, Haitham Khedr, Andrew Huang, et al. Sam 3: Segment anything with concepts. *arXiv preprint arXiv:2511.16719*, 2025.

[4] Mathilde Caron, Hugo Touvron, Ishan Misra, Hervé Jégou, Julien Mairal, Piotr Bojanowski, and Armand Joulin. Emerging properties in self-supervised vision transformers. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 9650–9660, 2021.

[5] 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*, 2019.

[6] Shoufa Chen, Chongjian Ge, Zhan Tong, Jiangliu Wang, Yibing Song, Jue Wang, and Ping Luo. Adaptformer: Adapting vision transformers for scalable visual recognition. *Advances in Neural Information Processing Systems*, 35:16664–16678, 2022.

[7] Sungha Choi, Sanghun Jung, Huiwon Yun, Joanne T Kim, Seungryong Kim, and Jaegul Choo. Robustnet: Improving domain generalization in urban-scene segmentation via instance selective whitening. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 11580–11590, 2021.

[8] Marius Cordts, Mohamed Omran, Sebastian Ramos, Timo Rehfeld, Markus Enzweiler, Rodrigo Benenson, Uwe Franke, Stefan Roth, and Bernt Schiele. The cityscapes dataset for semantic urban scene understanding. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 3213–3223, 2016.

[9] Muhammad Sohail Danish, Muhammad Haris Khan, Muhammad Akhtar Munir, M Saquib Sarfraz, and Mohsen Ali. Improving single domain-generalized object detection: A focus on diversification and alignment. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 17732–17742, 2024.

[10] Timothée Darcet, Maxime Oquab, Julien Mairal, and Piotr Bojanowski. Vision transformers need registers. *arXiv preprint arXiv:2309.16588*, 2023.

[11] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *2009 IEEE conference on computer vision and pattern recognition*, pages 248–255. Ieee, 2009.

[12] Jiankang Deng, Jia Guo, Niannan Xue, and Stefanos Zafeiriou. Arcface: Additive angular margin loss for deep face recognition. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 4690–4699, 2019.

[13] Wei Dong, Yuan Sun, Yiting Yang, Xing Zhang, Zhijun Lin, Qingsen Yan, Haokui Zhang, Peng Wang, Yang Yang, and Hengtao Shen. Efficient adaptation of pre-trained vision transformer via householder transformation. *Advances in Neural Information Processing Systems*, 37:102056–102077, 2024.

[14] Yuxin Fang, Quan Sun, Xinggang Wang, Tiejun Huang, Xinlong Wang, and Yue Cao. Eva-02: A visual representation for neon genesis. *Image and Vision Computing*, 149:105171, 2024.

[15] Yuxin Fang, Wen Wang, Binhui Xie, Quan Sun, Ledell Wu, Xinggang Wang, Tiejun Huang, Xinlong Wang, and Yue Cao. Eva: Exploring the limits of masked visual representation learning at scale. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 19358–19369, 2023.

[16] Robert Geirhos, Patricia Rubisch, Claudio Michaelis, Matthias Bethge, Felix A Wichmann, and Wieland Brendel. Imagenet-trained cnns are biased towards texture; increasing shape bias improves accuracy and robustness. In *International conference on learning representations*, 2018.

[17] Zeyu Han, Chao Gao, Jinyang Liu, Jeff Zhang, and Sai Qian Zhang. Parameter-efficient fine-tuning for large models: A comprehensive survey. *arXiv preprint arXiv:2403.14608*, 2024.

[18] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016.

[19] Dan Hendrycks, Norman Mu, Ekin D Cubuk, Barret Zoph, Justin Gilmer, and Balaji Lakshminarayanan. Augmix: A simple data processing method to improve robustness and uncertainty. *arXiv preprint arXiv:1912.02781*, 2019.

[20] Edward J Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Liang Wang, Weizhu Chen, et al. Lora: Low-rank adaptation of large language models. *Iclr*, 1(2):3, 2022.

[21] Lei Huang, Yi Zhou, Fan Zhu, Li Liu, and Ling Shao. Iterative normalization: Beyond standardization towards efficient whitening. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 4874–4883, 2019.

[22] Menglin Jia, Luming Tang, Bor-Chun Chen, Claire Cardie, Serge Belongie, Bharath Hariharan, and Ser-Nam Lim. Visual prompt tuning. In *European conference on computer vision*, pages 709–727. Springer, 2022.

[23] Han Jiang, Wenfei Yang, Tianzhu Zhang, and Yongdong Zhang. Diffusion-based source-biased model for single domain generalized object detection. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 1548–1557, 2025.

[24] Nick Jiang, Amil Dravid, Alexei Efros, and Yossi Gandelsman. Vision transformers don't need trained registers. *arXiv preprint arXiv:2506.08010*, 2025.

[25] Qing Jiang, Feng Li, Zhaoyang Zeng, Tianhe Ren, Shilong Liu, and Lei Zhang. T-rex2: Towards generic object detection via text-visual prompt synergy. In *European Conference on Computer Vision*, pages 38–57. Springer, 2024.

[26] Yujie Jin, Xu Chu, Yasha Wang, and Wenwu Zhu. Domain generalization through the lens of angular invariance. *arXiv preprint arXiv:2210.15836*, 2022.

[27] Prannay Khosla, Piotr Teterwak, Chen Wang, Aaron Sarna, Yonglong Tian, Phillip Isola, Aaron Maschinot, Ce Liu, and Dilip Krishnan. Supervised contrastive learning. *Advances in neural information processing systems*, 33:18661–18673, 2020.

[28] Alexander Kirillov, Eric Mintun, Nikhila Ravi, Hanzi Mao, Chloe Rolland, Laura Gustafson, Tete Xiao, Spencer Whitehead, Alexander C Berg, Wan-Yen Lo, et al. Segment anything. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 4015–4026, 2023.

[29] Wooju Lee, Dasol Hong, Hyungtae Lim, and Hyun Myung. Object-aware domain generalization for object detection. In *proceedings of the AAAI conference on artificial intelligence*, volume 38, pages 2947–2955, 2024.

[30] Chen Li, Huiying Xu, Changxin Gao, Zeyu Wang, Yun Liu, and Xinzhong Zhu. Towards single-source domain generalized object detection via causal visual prompts. *The Thirty-ninth Annual Conference on Neural Information Processing Systems*, 2025.

[31] Deng Li, Aming Wu, Yaowei Wang, and Yahong Han. Prompt-driven dynamic object-centric learning for single domain generalization. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 17606–17615, 2024.

[32] Chuang Lin, Zehuan Yuan, Sicheng Zhao, Peize Sun, Changhu Wang, and Jianfei Cai. Domain-invariant disentangled network for generalizable object detection. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 8771–8780, 2021.

[33] Weiyang Liu, Yandong Wen, Zhiding Yu, Ming Li, Bhiksha Raj, and Le Song. Sphereface: Deep hypersphere embedding for face recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 212–220, 2017.

[34] Yajing Liu, Shijun Zhou, Xiyao Liu, Chunhui Hao, Baojie Fan, and Jiandong Tian. Unbiased faster r-cnn for single-source domain generalized object detection. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 28838–28847, 2024.

[35] 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.

[36] Gerhard Neuhold, Tobias Ollmann, Samuel Rota Bulo, and Peter Kontschieder. The mapillary vistas dataset for semantic understanding of street scenes. In *Proceedings of the IEEE international conference on computer vision*, pages 4990–4999, 2017.

[37] Maxime Oquab, Timothée Darcet, Théo Moutakanni, Huy Vo, Marc Szafraniec, Vasil Khalidov, Pierre Fernandez, Daniel Haziza, Francisco Massa, Alaaeldin El-Nouby, et al. Dinov2: Learning robust visual features without supervision. *arXiv preprint arXiv:2304.07193*, 2023.

[38] Xingang Pan, Xiaohang Zhan, Jianping Shi, Xiaoou Tang, and Ping Luo. Switchable whitening for deep representation learning. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 1863–1871, 2019.

[39] Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, et al. Learning transferable visual models from natural language supervision. In *International conference on machine learning*, pages 8748–8763. PmLR, 2021.

[40] Nikhila Ravi, Valentin Gabeur, Yuan-Ting Hu, Ronghang Hu, Chaitanya Ryali, Tengyu Ma, Haitham Khedr, Roman Rädle, Chloe Rolland, Laura Gustafson, et al. Sam 2: Segment anything in images and videos. In *The Thirteenth International Conference on Learning Representations*, 2024.

[41] Shaoqing Ren, Kaiming He, Ross Girshick, and Jian Sun. Faster r-cnn: Towards real-time object detection with region proposal networks. *IEEE transactions on pattern analysis and machine intelligence*, 39(6):1137–1149, 2016.

[42] Stephan R Richter, Vibhav Vineet, Stefan Roth, and Vladlen Koltun. Playing for data: Ground truth from computer games. In *European conference on computer vision*, pages 102–118. Springer, 2016.

[43] Cheng Shi, Yizhou Yu, and Sibei Yang. Vision transformers need more than registers. *arXiv preprint arXiv:2602.22394*, 2026.

[44] Oriane Siméoni, Huy V Vo, Maximilian Seitzer, Federico Baldassarre, Maxime Oquab, Cijo Jose, Vasil Khalidov, Marc Szafraniec, Seungeun Yi, Michaël Ramamonjisoa, et al. Dinov3. *arXiv preprint arXiv:2508.10104*, 2025.

[45] Bo Sun, Banghuai Li, Shengcai Cai, Ye Yuan, and Chi Zhang. Fsce: Few-shot object detection via contrastive proposal encoding. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 7352–7362, 2021.

[46] Filip Szatkowski, Yaoyue Zheng, Fei Yang, BartŁ Twardowski, Joost van de Weijer, et al. Improving continual learning performance and efficiency with auxiliary classifiers. *arXiv preprint arXiv:2403.07404*, 2024.

[47] Vidit Vidit, Martin Engilberge, and Mathieu Salzmann. Clip the gap: A single domain generalization approach for object detection. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 3219–3229, 2023.

[48] Kunyu Wang, Xueyang Fu, Yuanfei Bao, Chengjie Ge, Chengzhi Cao, Wei Zhai, and Zheng-Jun Zha. Paid: Pairwise angular-invariant decomposition for continual test-time adaptation. *arXiv preprint arXiv:2506.02453*, 2025.

[49] Yuzhu Wang, Lechao Cheng, Chaowei Fang, Dingwen Zhang, Manni Duan, and Meng Wang. Revisiting the power of prompt for visual tuning. *arXiv preprint arXiv:2402.02382*, 2024.

[50] Zhixiang Wei, Lin Chen, Yi Jin, Xiaoxiao Ma, Tianle Liu, Pengyang Ling, Ben Wang, Huaian Chen, and Jinjin Zheng. Stronger fewer & superior: Harnessing vision foundation models for domain generalized semantic segmentation. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 28619–28630, 2024.

[51] Aming Wu and Cheng Deng. Single-domain generalized object detection in urban scene via cyclic-disentangled self-distillation. In *Proceedings of the IEEE/CVF Conference on computer vision and pattern recognition*, pages 847–856, 2022.

[52] Fan Wu, Jinling Gao, Lanqing Hong, Xinbing Wang, Chenghu Zhou, and Nanyang Ye. G-nas: Generalizable neural architecture search for single domain generalization object detection. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 38, pages 5958–5966, 2024.

[53] Lingling Xu, Haoran Xie, S Joe Qin, Xiaohui Tao, and Fu Lee Wang. Parameter-efficient fine-tuning methods for pretrained language models: A critical review and assessment. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2026.

[54] Xiaoran Xu, Jiangang Yang, Wenyue Chong, Wenhui Shi, Shichu Sun, Jing Xing, and Jian Liu. Boosting single-domain generalized object detection via vision-language knowledge interaction. In *Proceedings of the 33rd ACM International Conference on Multimedia*, pages 131–140, 2025.

[55] Xiaoran Xu, Jiangang Yang, Wenhui Shi, Siyuan Ding, Luqing Luo, and Jian Liu. Physaug: A physical-guided and frequency-based data augmentation for single-domain generalized object detection. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 39, pages 21815–21823, 2025.

[56] Fisher Yu, Haofeng Chen, Xin Wang, Wenqi Xian, Yingying Chen, Fangchen Liu, Vashisht Madhavan, and Trevor Darrell. Bdd100k: A diverse driving dataset for heterogeneous multitask learning. In *2020 IEEE/CVF conference on computer vision and pattern recognition (CVPR)*, pages 2633–2642. IEEE, 2020.

[57] Runyi Yu, Xiaoyu Chen, Hanyuan Zhang, Qi Zou, and Jianyong Guo. Semclip: enhancing domain generalization in object detection via style enhancement and semantic consistency: R. yu et al. *The Visual Computer*, 42(1):94, 2026.

[58] Shen Yuan, Haotian Liu, and Hongteng Xu. Bridging the gap between low-rank and orthogonal adaptation via householder reflection adaptation. *Advances in Neural Information Processing Systems*, 37:113484–113518, 2024.

[59] Seokju Yun, Seunghye Chae, Dongheon Lee, and Youngmin Ro. Soma: Singular value decomposed minor components adaptation for domain generalizable representation learning. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pages 25602–25612, 2025.

[60] Hao Zhang, Feng Li, Shilong Liu, Lei Zhang, Hang Su, Jun Zhu, Lionel M Ni, and Heung-Yeung Shum. Dino: Detr with improved denoising anchor boxes for end-to-end object detection. *arXiv preprint arXiv:2203.03605*, 2022.

[61] Yin Zhang, Yongqiang Zhang, Yaoyue Zheng, Bogdan Raducanu, and Dan Liu. Causaltune: Mining causal factors from vision foundation models for domain generalized semantic segmentation. *arXiv preprint arXiv:2512.16567*, 2025.

[62] Yaoyue Zheng, Yin Zhang, Joost van de Weijer, Gido M van de Ven, Shaoyi Du, Xuetao Zhang, and Zhiqiang Tian. Revisiting weight regularization for low-rank continual learning. *arXiv preprint arXiv:2602.17559*, 2026.

[63] Barret Zoph, Ekin D Cubuk, Golnaz Ghiasi, Tsung-Yi Lin, Jonathon Shlens, and Quoc V Le. Learning data augmentation strategies for object detection. In *European conference on computer vision*, pages 566–583. Springer, 2020.

Appendix

A Additional Ablation Studies and Analysis

A.1 Position of the HR Matrices within the Transformer Blocks.

In our proposed method, we introduce a learnable rotation matrix $\boldsymbol{R}$, defined as the product of r Householder Reflection (HR) matrices, and integrate it into the frozen Transformer blocks. We now investigate the effect of the insertion position of the HR matrices within the Transformer blocks on the S-DGOD task. Specifically, we conduct experiments on the QKV projection layer (QKV), the output projection layer ($Output$) and the feed-forward network (MLP layer, FFN) within the DINOv2 Transformer blocks.

As shown in Table 8, inserting HR matrices only into the QKV projection yields a substantial improvement over the baseline, increasing the mPC from 54.1% to 58.9% (first and second rows). This configuration consistently improves performance across all domains, indicating that enhancing the QKV projections strengthens the robustness of attention representations.

Further inserting HR matrices into both the QKV and $Output$ projection layers leads to the best overall performance, achieving the highest mPC of 59.4% and the best mAP (62.7%). In contrast, adding HR matrices to the FFN layer does not bring further gains and slightly degrades performance (mPC drops to 58.7%). This observation indicates that robustness improvements mainly originate from the attention layers rather than the feed-forward network. Consequently, we adopt the QKV + $Output$ configuration as our default setting.

Table 8: Position of the HR matrices within the Transformer blocks. First row is Baseline. QKV, $Output$, and FFN denote QKV projection layer, output projection layer and feed-forward network (MLP layer), respectively.

QKV	$Output$	FFN	DS.	DF.	DR.	NR.	NS.	Avg.	**mPC**
-	-	-	71.2	53.5	60.8	42.6	59.5	57.5	54.1
✓	-	-	<u>75.1</u>	<u>57.5</u>	65.5	**50.1**	62.3	<u>62.1</u>	<u>58.9</u>
✓	✓	-	**76.0**	**58.3**	**65.9**	<u>49.9</u>	**63.3**	**62.7**	**59.4**
✓	✓	✓	75.0	<u>57.5</u>	<u>65.6</u>	49.4	<u>62.4</u>	62.0	58.7

A.2 Performance on Domain Generalized Semantic Segmentation.

We additionally evaluate GIFT on the domain generalized semantic segmentation (DGSS) task. We use Rein [50] as baseline and following previous methods [50, 2, 59, 61], we compare our method with existing DGSS methods on two setups: (1) Cityscapes (*C.*) [8] → BDD100K (*B.*) [56], Mapillary (*M.*) [36] and (2) GTA5 (*G.*) [42] → (*C.*), (*B.*), (*M.*). As shown in Table 9, our method achieves the best results in 3 out of 5 cases and the second-best results in 1 case compared to VFM-based methods. These results demonstrate that the proposed geometry-preserving fine-tuning strategy generalizes beyond object detection and is also effective for domain generalized semantic segmentation.

Table 9: DGSS performance of our proposed GIFT. (*C.*), (*B.*), (*M.*), and (*G.*) denote Cityscapes, BDD100K, Mapillary and GTA5 datasets, respectively.

Methods	*C.* → *B.*	*C.* → *M.*	*G.* → *C.*	*G.* → *B.*	*G.* → *M.*
Rein [50]	63.54	74.03	66.40	60.40	66.10
FADA [2]	65.12	75.86	68.23	<u>61.94</u>	68.09
SoMA [59]	<u>67.02</u>	<u>76.45</u>	**71.82**	61.31	**71.67**
Causal-Tune [61]	66.28	76.05	66.22	61.80	<u>68.21</u>
Ours	**67.42**	**76.71**	<u>68.60</u>	**62.01**	68.08

A.3 Performance across Different VFMs.

We further validate the proposed method using three different VFMs, *i.e.*, EVA02-Large [15, 14], SAM-Huge [28], and DINOv2-Large [37]. As shown in Table 10, our method achieves strong performance across all three VFM backbones, demonstrating strong generalization. Specifically, our

Table 10: Performance of different VFMs with a DETR-based [60] detection head. "Avg." indicates the mean mAP across the five conditions.

VFMs	Fine-tune Method	DS.	DF.	DR.	NR.	NS.	Avg.	**mPC**
EVA02-Large [15, 14]	Freeze	63.0	45.2	48.6	27.3	38.8	44.6	40.0
	+ Rein [50]	68.3	49.2	54.8	32.2	48.1	50.5(+5.9↑)	46.1(+6.1↑)
	+ Cauvis [30]	69.7	50.2	57.6	34.2	48.1	52.0(+7.4↑)	47.5(+7.5↑)
	+ Ours	**71.6**	**51.3**	**59.2**	**35.9**	**49.2**	**53.4**(+8.8↑)	**48.9**(+8.9↑)
SAM-Huge [28]	Freeze	69.7	50.5	52.5	28.8	52.9	50.9	46.2
	+ Rein [50]	70.0	51.9	54.0	30.9	54.4	52.2(+1.3↑)	47.8(+1.6↑)
	+ Cauvis [30]	72.2	**53.7**	55.8	31.9	**55.7**	53.9(+3.0↑)	49.3(+3.1↑)
	+ Ours	**72.8**	53.6	**56.2**	**33.1**	**55.7**	**54.3**(+3.4↑)	**49.7**(+3.5↑)
DINOv2-Large [37]	Freeze	71.2	53.5	60.8	42.6	59.5	57.5	54.1
	+ Rein [50]	72.8	55.0	62.4	45.2	59.4	59.0(+1.5↑)	55.5(+1.4↑)
	+ Cauvis [30]	73.7	56.5	64.6	47.6	61.2	60.7(+3.2↑)	57.5(+3.4↑)
	+ Ours	**76.0**	**58.3**	**65.9**	**49.9**	**63.3**	**62.7**(+5.2↑)	**59.4**(+5.3↑)

method achieves 48.9% mPC on the EVA02 [14] backbone, surpassing frozen EVA02 by a large margin (+8.9%). Moreover, it brings a +5.3% improvement on the DINOv2 [37] backbone and achieves a satisfactory performance (49.7% mPC) on SAM [28]. In addition, our method still has significant advantages over the current best method Cauvis [30], achieving improvements of +1.4%, +0.4%, and +1.9% on EVA02, SAM and DINOv2 backbones, respectively. These results indicate that our method enhances the performance of different VFMs and provides stable improvements across diverse weather conditions, highlighting its effectiveness for S-DGOD tasks.

B More Implementation Details

Our environment is Ubuntu 22.04, Python 3.9.24, CUDA 11.8, PyTorch 2.2.0, and MMDetection 3.3.0. We adopt the data augmentation pipeline provided by MMDetection and use only the basic augmentation strategy from the original detector for fair comparison.

For DWD [51], following the standard protocol, Daytime-Sunny is treated as the source domain, containing 18205 training images and 8313 evaluation images. The Daytime-Foggy, Dusk-Rainy, Night-Rainy, and Night-Sunny conditions serve as unseen target domains, with 26158, 2494, 3501, and 3775 images, respectively. For Cityscapes-C [35], we use the clean Cityscapes [8] training set as the source domain, containing 2975 training images, while the corrupted images serve as unseen target domains for evaluation.

For more details on the Feature Correlation Analysis in Section 4.3, we extract object-level features from the penultimate layer of the classification head and group them according to their object categories. Given an input image, the Transformer backbone produces a sequence of patch tokens. We then compute the average cosine similarity between class-wise features from the clean and corrupted domains to construct the feature correlation matrix.

C Detailed Experimental Results

C.1 Detailed Results on Diverse Weather Dataset.

Following Cauvis [30], we group the 7 categories in DWD [51] into three groups: Heavy Vehicles '$hev.$' (Bus, Truck), Mid-sized Vehicles '$mid.$' (Car, Motor), and Non-motorized Vehicles '$non.$' (Bike, Person, Rider). 'all' denotes the mAP computed over all categories under the given weather condition.

As shown in Table 11, under diverse weather-induced domain shifts, the proposed method consistently achieves the best performance across four unseen target domains. In particular, GIFT attains the highest mPC of 57.0%, surpassing the previous SOTA method, Cauvis [30], by 2.9%, and outperforming the baseline (Frozen DINOv2 [37] with Faster R-CNN [41]) by 4.9%. In addition, GIFT demonstrates a significant performance margin over traditional S-DGOD methods such as UFR [34](+23.8%), PDD [31](+24.4%), and G-NAS [52](+23.5%), showing that the proposed VFM-based fine-tuning strategy effectively addresses the main bottleneck in S-DGOD.

Table 11: Detailed results on Diverse Weather Dataset [51], using Faster R-CNN as the detection head. mPC denotes the average performance across 4 different adverse weather conditions. **Bold** and underline results represent best and second-best performance, respectively. **GP** denotes geometric preservation.

	Daytime-Foggy				Dusk-Rainy				Night-Rainy				Night-Sunny				
Methods	*hev.*	*mid.*	*non.*	***all***	*hev.*	*mid.*	*non.*	***all***	*hev.*	*mid.*	*non.*	***all***	*hev.*	*mid.*	*non.*	***all***	**mPC**
IN [21]	25.7	33.4	27.0	28.5	34.3	25.0	13.7	22.8	19.5	11.5	8.8	12.6	38.3	27.4	25.3	29.6	23.4
SW [38]	27.1	34.9	33.8	32.2	37.0	30.3	16.6	26.3	20.0	13.9	9.4	13.7	39.5	33.2	29.6	33.4	26.1
FRCNN [41]	30.6	37.8	32.6	33.5	34.6	32.0	21.1	28.0	18.7	17.5	9.0	14.2	43.3	33.7	32.2	35.8	27.9
CDSD [51]	28.5	39.3	32.9	33.5	38.9	32.2	18.5	28.2	21.8	19.7	11.2	16.6	42.0	35.2	34.0	36.6	28.7
ClipGap [47]	30.7	46.0	38.9	38.5	40.1	38.8	22.8	32.3	25.8	22.7	11.3	18.7	40.3	38.6	33.5	36.9	31.6
G-NAS [52]	28.5	44.8	36.1	36.4	45.2	40.6	24.7	35.1	25.0	19.3	11.1	17.4	47.4	47.0	42.2	45.0	33.5
PDD [31]	31.2	45.9	39.7	39.1	41.7	40.7	23.9	33.7	24.3	23.0	13.1	19.2	41.9	40.2	35.1	38.5	32.6
UFR [34]	32.2	47.7	39.2	39.6	40.5	42.2	22.4	33.2	26.6	22.8	11.8	19.2	45.6	40.4	37.9	40.8	33.2
Baseline	51.0	56.0	46.8	50.6	63.9	62.6	53.4	59.0	55.9	44.2	34.6	43.4	59.4	56.2	52.3	55.4	52.1
Cauvis [30]	50.5	58.0	49.6	52.3	67.0	66.8	53.9	61.3	55.2	46.1	35.3	44.1	60.9	58.7	55.7	58.3	54.1
GP Only	51.0	**61.0**	**53.9**	**55.1**	**68.8**	66.1	**56.7**	62.8	57.7	**51.2**	36.6	46.6	62.8	61.1	59.5	60.9	56.4
GIFT	**51.7**	60.7	53.3	54.9	68.7	**68.1**	**59.1**	**64.4**	**59.0**	50.8	**36.8**	**47.1**	**62.9**	**61.7**	**61.0**	**61.7**	**57.0**

Table 12: Detailed results on Daytime-Sunny (source domain) of DWD. Baseline is a frozen DINOv2 [37] backbone with a Faster R-CNN [41] detector. 'F' and 'D' denote Faster R-CNN and DETR-based [60] detector.

	Daytime-Sunny							
Methods	Bus	Bike	Car	Motor	Person	Rider	Truck	**mAP**
InterNorm [21]	58.4	34.2	42.2	44.1	31.6	40.8	55.5	43.9
SW [38]	62.3	42.9	53.3	49.9	39.2	46.2	60.6	50.6
ISW [7]	62.9	44.6	53.5	49.2	39.9	48.3	60.9	51.3
Faster R-CNN [41]	66.9	45.9	69.8	46.5	50.6	49.6	64.0	56.2
CDSD [51]	68.8	50.9	53.9	56.2	41.8	52.4	68.7	56.1
ClipGap [47]	55.0	47.8	67.5	46.7	49.4	46.7	54.7	52.5
UFR [34]	66.8	51.0	70.6	55.8	49.8	48.5	67.4	58.6
Cauvis [30] (D)	76.3	64.3	87.7	64.2	79.2	72.1	71.8	73.7
Ours (F)	77.3	61.3	89.1	63.8	77.5	76.0	72.4	73.9
Ours (D)	**77.6**	**67.0**	87.5	**68.9**	**80.7**	**76.3**	**74.1**	**76.0**

Notably, even when only geometric preservation is applied, the model still outperforms previous methods. Across the four unseen domains, it improves over the baseline by 4.5%, 3.8%, 3.2%, and 5.5%, and surpasses the SOTA method Cauvis [30] by 2.8%, 1.5%, 2.5%, and 2.6%, highlighting the effectiveness of preserving the geometric structure of pre-trained representations.

C.2 Results on Daytime-Sunny.

We report the detection results on the source domain of DWD [51]. As shown in Table 12, our method significantly outperforms traditional S-DGOD methods in terms of mAP. For example, when using the same detector, Faster R-CNN [41], our method achieves improvements of +15.3% over UFR [34] (from 58.6% to 73.9%) and +21.4% over ClipGap [47] (52.5% *vs.* 73.9%). Surprisingly, our method with Faster R-CNN detector is even superior to Cauvis [30], which uses a DETR-based detector (73.9% *vs.* 73.7%). Moreover, when using a DETR-based detector, our method achieves a new state-of-the-art result of 76% in mAP, surpassing Cauvis by 2.3%, and achieves the best performance in 6 out of the 7 categories.

C.3 Results on Unseen Target Domain.

Table 13 presents a comprehensive evaluation of our proposed method compared with several S-DGOD methods across four unseen target domains in DWD [51].

Daytime-Foggy. Our DETR-based model (Ours (D)) achieves the best overall performance with 58.3% mAP, surpassing the previous state-of-the-art method Cauvis [30] by +1.8% mAP. Specifically, our method consistently improves most object categories, including Bus (55.0% *vs.* 50.7%), Bike (46.1% *vs.* 43.8%), Motor (51.7% *vs.* 50.5%), Person (68.8% *vs.* 67.4%), and Rider (64.3% *vs.* 61.9%). Notably, our Faster R-CNN-based version (Ours (F)) also performs competitively,

Table 13: Detailed results on unseen target domain of DWD. 'Mot', 'Per.', 'Rid.', and 'Tru.' denote 'Motor', 'Person', 'Rider', and 'Truck', respectively. 'F' and 'D' denote Faster R-CNN [41] and DETR-based [60] detector.

	Daytime-Foggy								Dusk-Rainy							
Methods	Bus	Bike	Car	Mot.	Per.	Rid.	Tru.	**mAP**	Bus	Bike	Car	Mot.	Per.	Rid.	Tru.	**mAP**
FR [41]	34.5	29.6	49.3	26.2	33.0	35.1	26.7	33.5	34.2	21.8	47.9	16.0	22.9	18.5	34.9	28.0
CDSD [51]	32.9	28.0	48.8	29.8	32.5	38.2	24.1	33.5	37.1	19.6	50.9	13.4	19.7	16.3	40.7	28.2
ClipGap [47]	36.2	34.2	57.9	34.0	38.7	43.8	25.1	38.5	37.8	22.8	60.7	16.8	26.8	18.7	42.4	32.3
G-NAS [52]	32.4	31.2	57.7	31.9	38.6	38.5	24.5	36.4	44.6	22.3	66.4	14.7	32.1	19.6	45.8	35.1
PDD [31]	36.1	34.5	58.4	33.3	40.5	44.2	26.2	39.1	39.4	25.2	60.9	20.4	29.9	16.5	43.9	33.7
UFR [34]	36.9	35.8	61.7	33.7	39.5	42.2	27.5	39.6	37.1	21.8	67.9	16.4	27.4	17.9	43.9	33.2
Cauvis [30] (D)	50.7	43.8	73.8	50.5	67.4	61.9	47.5	56.5	67.2	54.8	85.6	52.0	65.5	57.2	70.1	64.6
Ours (F)	54.2	38.7	73.9	47.4	61.9	59.3	**49.1**	54.9	67.3	51.8	**87.1**	49.0	65.4	**60.2**	70.0	64.4
Ours (D)	**55.0**	**46.1**	**74.1**	**51.7**	**68.8**	**64.3**	48.0	**58.3**	**68.5**	**55.2**	85.7	**57.9**	**66.8**	56.4	**71.0**	**65.9**

	Night-Rainy								Night-Sunny							
Methods	Bus	Bike	Car	Mot.	Per.	Rid.	Tru.	**mAP**	Bus	Bike	Car	Mot.	Per.	Rid.	Tru.	**mAP**
FR [41]	21.3	7.7	28.8	6.1	8.9	10.3	16.0	14.2	43.5	31.2	49.8	17.5	36.3	29.2	43.1	35.8
CDSD [51]	24.4	11.6	29.5	9.8	10.5	11.4	19.2	16.6	40.6	35.1	50.7	19.7	34.7	32.1	43.4	36.6
ClipGap [47]	28.6	12.1	36.1	9.2	12.3	9.6	22.9	18.7	37.7	34.3	58.0	19.2	37.6	28.5	42.9	36.9
G-NAS [52]	28.6	9.8	38.4	0.1	13.8	9.8	21.4	17.4	46.9	40.5	67.5	26.5	50.7	35.4	47.8	45.0
PDD [31]	25.6	12.1	35.8	10.1	14.2	12.9	22.9	19.2	40.9	35.0	59.0	21.3	40.4	29.9	42.9	38.5
UFR [34]	29.9	11.8	36.1	9.4	13.1	10.5	23.3	19.2	43.6	38.1	66.1	14.7	49.1	26.4	47.5	40.8
Cauvis [30] (D)	60.8	32.4	69.5	24.0	49.9	**39.5**	57.2	47.6	62.8	55.1	79.5	39.3	70.4	57.4	64.2	61.2
Ours (F)	61.5	29.5	**73.0**	**28.6**	48.8	32.2	56.5	47.1	62.8	53.9	**83.5**	39.8	66.8	**62.3**	63.0	61.7
Ours (D)	**64.7**	**38.5**	70.9	27.1	**50.0**	37.3	**61.0**	**49.9**	**63.9**	**59.3**	80.5	**40.5**	**72.6**	60.6	**65.5**	**63.3**

particularly in the Truck category, where it achieves the highest accuracy of 49.1%. Despite the reduced visibility caused by fog, our method still demonstrates strong robustness.

Dusk-Rainy. Our method achieves 65.9% mAP, outperforming Cauvis [30] by +1.3% mAP. It is worth noting that Ours (F) exhibits exceptional performance in the Car category, reaching 87.1% mAP, and the Rider category at 60.2%. These results suggest that our framework effectively preserves geometric structure even when visual clarity is compromised by rain artifacts and low-light conditions during dusk.

Night-Rainy. The Night-Rainy domain is the most challenging scenario due to extremely low illumination and rain artifacts. Although all models experience performance degradation, Ours (D) maintains a clear lead with 49.9% mAP. In particular, the improvement on Bike (from 32.4% to 38.5%, +6.1% mAP) demonstrates the effectiveness of our method in detecting small objects under severe visual degradation.

Night-Sunny. Our models continue to dominate the results under Night-Sunny conditions. Ours (D) achieves the highest mAP of 63.3%, showing superior performance in the Person (72.6%) and Bike (59.3%) categories. These results demonstrate that our method effectively handles object appearance variations caused by strong illumination contrast in nighttime scenes.

Overall, the consistent improvements across all domains demonstrate that our method effectively enhances cross-domain robustness and generalization under diverse adverse weather and illumination conditions.

C.4 Results on Cityscapes-C from Level-1 to Level-5.

Fig. 10 and Table 14 present a detailed comparison between our method and Cauvis [30] on the Cityscapes-C [35] dataset across 5 severity levels. As shown in Fig. 10, our method consistently outperforms Cauvis across almost all corruption types and severity levels. Specifically, our method achieves higher mAP under both Noise-based corruptions (e.g., Gaussian, Shot, and Impulse) and Blur-based corruptions (e.g., Defocus, Glass, Motion, and Zoom), demonstrating stronger robustness to various perturbations.

As the corruption severity increases from Level-1 to Level-5, the performance of both methods gradually decreases. However, our method exhibits a slower degradation trend compared to Cauvis. For instance, under severe corruptions such as Level-5, our method still maintains noticeably higher results across most corruption types. Moreover, as shown in Table 14, our method achieves an excellent performance (43.5% in mPC) compared with Cauvis (35.6%), showing an overall improvement

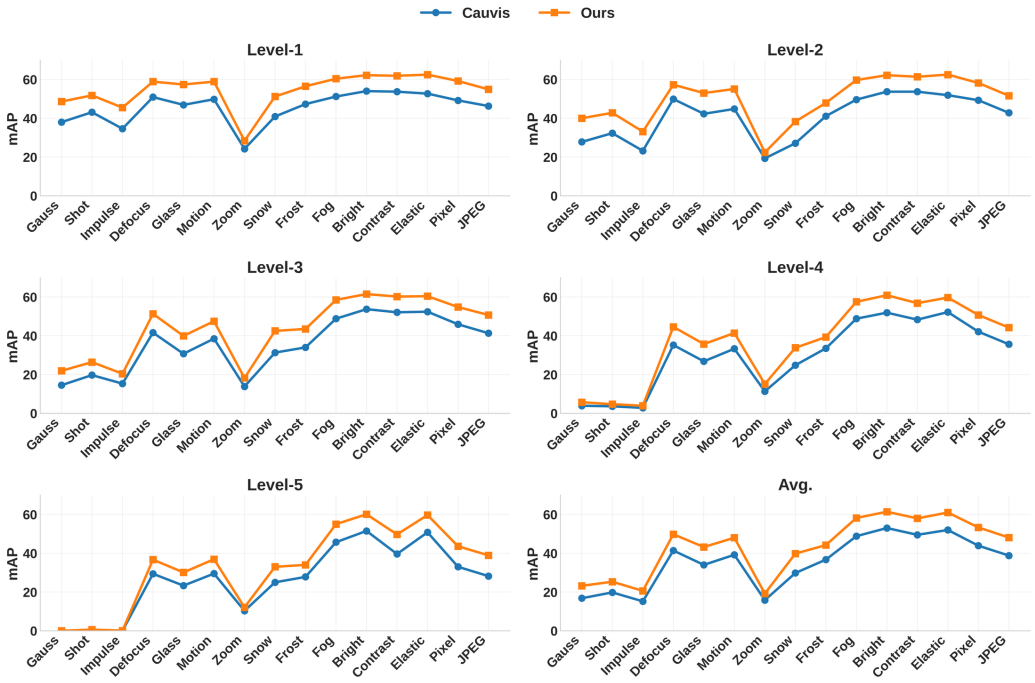

Figure 10: Comparison between our method and Cauvis [30] on Cityscapes-C from Level-1 to Level-5. 'Avg.' denotes the average mAP across the 5 corruption levels.

Table 14: Quantitative results on Cityscapes-C from Level-1 to Level-5. 'Avg.' denotes the average mAP across the 5 corruption levels.

	GIFT (Ours)						Cauvis [30]					
Types	Level-1	Level-2	Level-3	Level-4	Level-5	Avg.	Level-1	Level-2	Level-3	Level-4	Level-5	Avg.
Source	-	-	-	-	-	62.7	-	-	-	-	-	54.6
Gauss	48.6	40.0	21.9	5.7	0.0	23.2	38.0	27.8	14.5	3.9	0.0	16.8
Shot	51.2	42.8	26.4	4.7	0.6	25.1	43.1	32.3	19.7	3.6	0.1	19.8
Impulse	45.5	33.1	20.4	3.9	0.1	20.6	34.6	23.2	15.3	2.8	0.0	15.2
Defocus	58.9	57.3	51.3	44.6	36.7	49.8	50.9	49.9	41.6	35.2	29.4	41.4
Glass	57.4	53.0	39.9	35.7	30.1	43.2	46.9	42.3	30.7	26.8	23.3	34.0
Motion	58.9	55.1	47.5	41.4	36.9	48.0	49.8	44.8	38.5	33.3	29.5	39.2
Zoom	28.4	22.4	18.2	15.1	12.1	19.2	24.2	19.3	13.8	11.3	10.3	15.8
Snow	51.2	38.3	42.5	33.8	33.0	39.8	40.9	27.1	31.3	24.8	25.0	29.8
Frost	56.5	47.9	43.5	39.3	34.0	44.2	47.3	41.1	34.0	33.5	27.8	36.7
Fog	60.4	59.7	58.5	57.5	55.0	58.2	51.2	49.6	48.8	48.8	45.7	48.8
Bright	62.2	62.2	61.5	60.9	60.1	61.4	54.0	53.7	53.7	51.9	51.5	53.0
Contrast	61.9	61.4	60.2	56.8	49.7	58.0	53.7	53.7	52.1	48.3	39.6	49.5
Elastic	62.5	62.5	60.4	59.7	59.7	61.0	52.7	51.9	52.4	52.2	50.8	52.0
Pixel	59.2	58.2	54.8	50.7	43.6	53.3	49.2	49.3	45.9	42.1	33.0	43.9
JPEG	54.9	51.6	50.7	44.2	38.9	48.1	46.3	42.8	41.3	35.6	28.2	38.8
mPC	-	-	-	-	-	**43.5**	-	-	-	-	-	35.6

of +7.9%. These results demonstrate that our method substantially improves the robustness of the model against diverse corruptions.

D More Visualization of Detection Results

To further demonstrate the robustness of our method, we provide more visualization of detection results across four challenging target domains: Daytime-Foggy, Dusk-Rainy, Night-Rainy, and Night-Sunny in Fig. 11.

Daytime-Foggy. In foggy conditions characterized by low contrast and obscured details, our method demonstrates superior capability in identifying distant cars and trucks that are often incorrectly detected by ClipGap [47] and the Baseline. Moreover, the Baseline and Cauvis [30] often produce

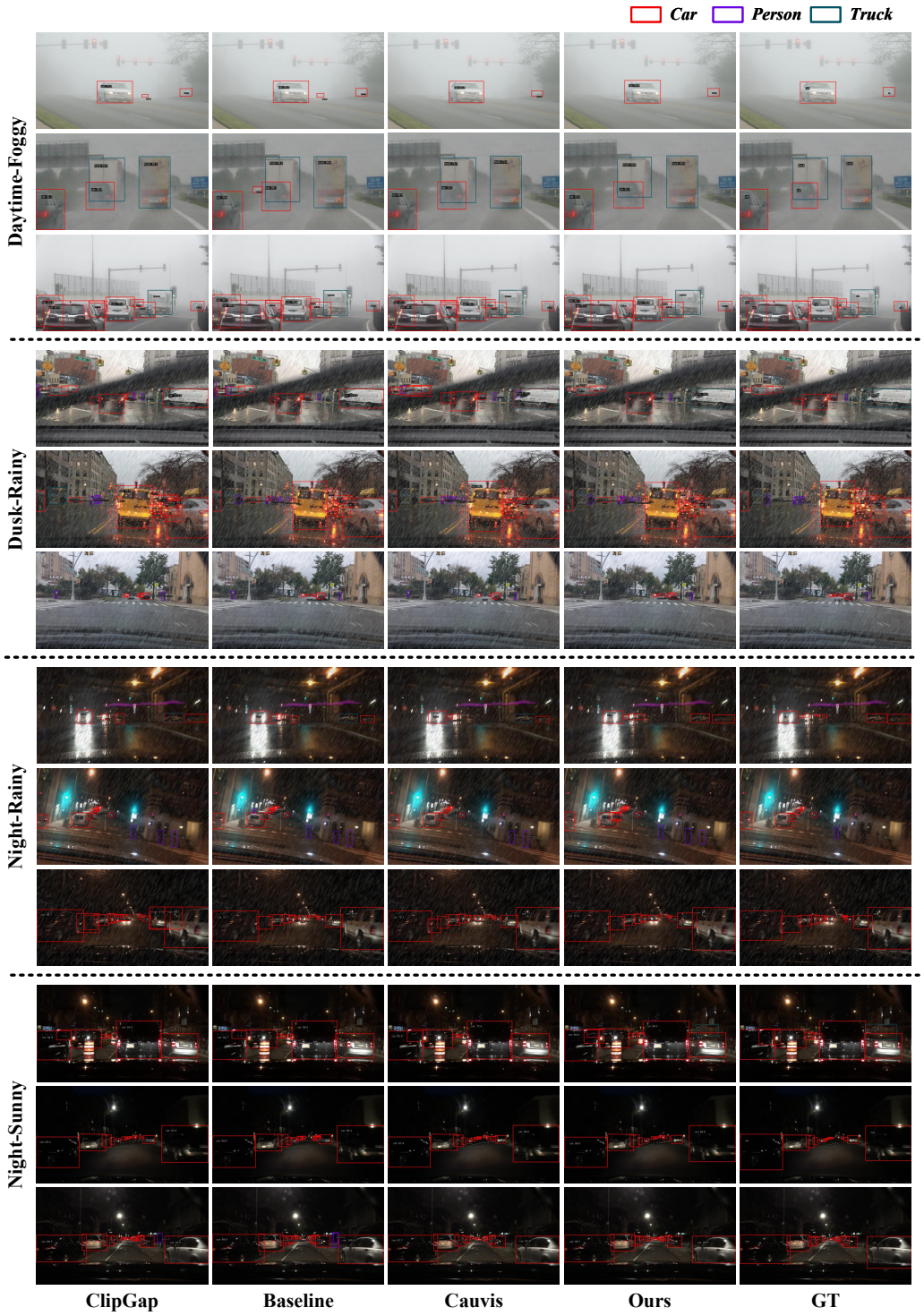

Figure 11: More visualization of detection results on the DWD [51]. (a) to (d) correspond to the four weather conditions of the DWD target domain: DF., DR., NR., and NS. GIFT is compared with the traditional S-DGOD method ClipGap [47], Baseline (Frozen DINOv2 [37] with Faster R-CNN [41]) and the SOTA method Cauvis [30]. Best seen on computer, in color and zoomed in.

inaccurate bounding boxes due to the degraded visibility while our model remains well aligned with the ground truth (GT).

Dusk-Rainy. In this scenario, other models frequently produce incomplete detections or inaccurate localization due to reduced visibility and illumination variations. In contrast, our method accurately detects persons and trucks despite the visual noise caused by rain streaks, indicating improved robustness to simultaneous illumination and weather variations.

Night-Rainy. In this setting, most other methods suffer from severe detection failures. For instance, ClipGap [47], Baseline and Cauvis [30] mistakenly detect cars as trucks. In contrast, our model shows remarkable resilience against glare from streetlights and headlights. These results demonstrate that the proposed method can better preserve geometric structure even under extreme nighttime weather conditions.

Night-Sunny. Although this domain does not contain rain or fog, extremely low illumination still poses significant challenges for object detection. Many other methods fail to detect small cars or distant trucks while our predictions are nearly indistinguishable from the GT.

E Limitation

First, when the pre-trained geometry is not well aligned with the downstream task, the relative improvement may be smaller than that observed in natural RGB driving scenarios, such as infrared or remote sensing images that differ substantially from natural-image pretraining data. Second, although GIFT reduces the number of trainable parameters and improves adaptation efficiency, its reliance on large-scale pre-trained VFMs still requires considerable GPU memory. In addition, this work focuses on a fixed category set. Its effectiveness under open-vocabulary domain generalization, where the detector needs to generalize to both unseen domains and novel categories, remains unexplored. Extending the proposed framework to open-vocabulary domain generalization is an important direction for future work.

NeurIPS Paper Checklist

1. **Claims**

   Question: Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope?

   Answer: [Yes]

   Justification: We state our main contributions in the abstract and Section 1.

   Guidelines:

   - The answer [N/A] means that the abstract and introduction do not include the claims made in the paper.
   - The abstract and/or introduction should clearly state the claims made, including the contributions made in the paper and important assumptions and limitations. A [No] or [N/A] answer to this question will not be perceived well by the reviewers.
   - The claims made should match theoretical and experimental results, and reflect how much the results can be expected to generalize to other settings.
   - It is fine to include aspirational goals as motivation as long as it is clear that these goals are not attained by the paper.

2. **Limitations**

   Question: Does the paper discuss the limitations of the work performed by the authors?

   Answer: [Yes]

   Justification: We discuss the limitation in Appendix E

   Guidelines:

   - The answer [N/A] means that the paper has no limitation while the answer [No] means that the paper has limitations, but those are not discussed in the paper.
   - The authors are encouraged to create a separate "Limitations" section in their paper.
   - The paper should point out any strong assumptions and how robust the results are to violations of these assumptions (e.g., independence assumptions, noiseless settings, model well-specification, asymptotic approximations only holding locally). The authors should reflect on how these assumptions might be violated in practice and what the implications would be.
   - The authors should reflect on the scope of the claims made, e.g., if the approach was only tested on a few datasets or with a few runs. In general, empirical results often depend on implicit assumptions, which should be articulated.
   - The authors should reflect on the factors that influence the performance of the approach. For example, a facial recognition algorithm may perform poorly when image resolution is low or images are taken in low lighting. Or a speech-to-text system might not be used reliably to provide closed captions for online lectures because it fails to handle technical jargon.
   - The authors should discuss the computational efficiency of the proposed algorithms and how they scale with dataset size.
   - If applicable, the authors should discuss possible limitations of their approach to address problems of privacy and fairness.
   - While the authors might fear that complete honesty about limitations might be used by reviewers as grounds for rejection, a worse outcome might be that reviewers discover limitations that aren't acknowledged in the paper. The authors should use their best judgment and recognize that individual actions in favor of transparency play an important role in developing norms that preserve the integrity of the community. Reviewers will be specifically instructed to not penalize honesty concerning limitations.

3. **Theory assumptions and proofs**

   Question: For each theoretical result, does the paper provide the full set of assumptions and a complete (and correct) proof?

   Answer: [Yes]

Justification: All theoretical results are presented with clearly stated assumptions, please check Section 3.2 and Section 3.3 for proof.

Guidelines:

- The answer [N/A] means that the paper does not include theoretical results.
- All the theorems, formulas, and proofs in the paper should be numbered and cross-referenced.
- All assumptions should be clearly stated or referenced in the statement of any theorems.
- The proofs can either appear in the main paper or the supplemental material, but if they appear in the supplemental material, the authors are encouraged to provide a short proof sketch to provide intuition.
- Inversely, any informal proof provided in the core of the paper should be complemented by formal proofs provided in appendix or supplemental material.
- Theorems and Lemmas that the proof relies upon should be properly referenced.

4. **Experimental result reproducibility**

   Question: Does the paper fully disclose all the information needed to reproduce the main experimental results of the paper to the extent that it affects the main claims and/or conclusions of the paper (regardless of whether the code and data are provided or not)?

   Answer: [Yes]

   Justification: We have described implementation details in Section 4.1 and Appendix B. Additionally, the code is provided in the supplementary materials.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - If the paper includes experiments, a [No] answer to this question will not be perceived well by the reviewers: Making the paper reproducible is important, regardless of whether the code and data are provided or not.
   - If the contribution is a dataset and/or model, the authors should describe the steps taken to make their results reproducible or verifiable.
   - Depending on the contribution, reproducibility can be accomplished in various ways. For example, if the contribution is a novel architecture, describing the architecture fully might suffice, or if the contribution is a specific model and empirical evaluation, it may be necessary to either make it possible for others to replicate the model with the same dataset, or provide access to the model. In general. releasing code and data is often one good way to accomplish this, but reproducibility can also be provided via detailed instructions for how to replicate the results, access to a hosted model (e.g., in the case of a large language model), releasing of a model checkpoint, or other means that are appropriate to the research performed.
   - While NeurIPS does not require releasing code, the conference does require all submissions to provide some reasonable avenue for reproducibility, which may depend on the nature of the contribution. For example
     (a) If the contribution is primarily a new algorithm, the paper should make it clear how to reproduce that algorithm.
     (b) If the contribution is primarily a new model architecture, the paper should describe the architecture clearly and fully.
     (c) If the contribution is a new model (e.g., a large language model), then there should either be a way to access this model for reproducing the results or a way to reproduce the model (e.g., with an open-source dataset or instructions for how to construct the dataset).
     (d) We recognize that reproducibility may be tricky in some cases, in which case authors are welcome to describe the particular way they provide for reproducibility. In the case of closed-source models, it may be that access to the model is limited in some way (e.g., to registered users), but it should be possible for other researchers to have some path to reproducing or verifying the results.

5. **Open access to data and code**

Question: Does the paper provide open access to the data and code, with sufficient instructions to faithfully reproduce the main experimental results, as described in supplemental material?

Answer: [Yes]

Justification: The datasets are public and open-access, please refer to Section 4.1 for details. Furthermore, the code and data access instructions are provided in the supplementary materials, enabling reproduction of the main results.

Guidelines:

- The answer [N/A] means that paper does not include experiments requiring code.
- Please see the NeurIPS code and data submission guidelines (`https://neurips.cc/public/guides/CodeSubmissionPolicy`) for more details.
- While we encourage the release of code and data, we understand that this might not be possible, so [No] is an acceptable answer. Papers cannot be rejected simply for not including code, unless this is central to the contribution (e.g., for a new open-source benchmark).
- The instructions should contain the exact command and environment needed to run to reproduce the results. See the NeurIPS code and data submission guidelines (`https://neurips.cc/public/guides/CodeSubmissionPolicy`) for more details.
- The authors should provide instructions on data access and preparation, including how to access the raw data, preprocessed data, intermediate data, and generated data, etc.
- The authors should provide scripts to reproduce all experimental results for the new proposed method and baselines. If only a subset of experiments are reproducible, they should state which ones are omitted from the script and why.
- At submission time, to preserve anonymity, the authors should release anonymized versions (if applicable).
- Providing as much information as possible in supplemental material (appended to the paper) is recommended, but including URLs to data and code is permitted.

6. **Experimental setting/details**

Question: Does the paper specify all the training and test details (e.g., data splits, hyperparameters, how they were chosen, type of optimizer) necessary to understand the results?

Answer: [Yes]

Justification: Detailed experimental setting/details are presented in Section 4.1 and Appendix B.

Guidelines:

- The answer [N/A] means that the paper does not include experiments.
- The experimental setting should be presented in the core of the paper to a level of detail that is necessary to appreciate the results and make sense of them.
- The full details can be provided either with the code, in appendix, or as supplemental material.

7. **Experiment statistical significance**

Question: Does the paper report error bars suitably and correctly defined or other appropriate information about the statistical significance of the experiments?

Answer: [No]

Justification: In the field of this paper and the evaluation protocols of these S-DGOD datasets do not require reporting the error bar.

Guidelines:

- The answer [N/A] means that the paper does not include experiments.
- The authors should answer [Yes] if the results are accompanied by error bars, confidence intervals, or statistical significance tests, at least for the experiments that support the main claims of the paper.

- The factors of variability that the error bars are capturing should be clearly stated (for example, train/test split, initialization, random drawing of some parameter, or overall run with given experimental conditions).
- The method for calculating the error bars should be explained (closed form formula, call to a library function, bootstrap, etc.)
- The assumptions made should be given (e.g., Normally distributed errors).
- It should be clear whether the error bar is the standard deviation or the standard error of the mean.
- It is OK to report 1-sigma error bars, but one should state it. The authors should preferably report a 2-sigma error bar than state that they have a 96% CI, if the hypothesis of Normality of errors is not verified.
- For asymmetric distributions, the authors should be careful not to show in tables or figures symmetric error bars that would yield results that are out of range (e.g., negative error rates).
- If error bars are reported in tables or plots, the authors should explain in the text how they were calculated and reference the corresponding figures or tables in the text.

8. **Experiments compute resources**

   Question: For each experiment, does the paper provide sufficient information on the computer resources (type of compute workers, memory, time of execution) needed to reproduce the experiments?

   Answer: [Yes]

   Justification: We have provided information about the computational resources used for experiments, including the type of GPUs and execution details in Section 4.1.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - The paper should indicate the type of compute workers CPU or GPU, internal cluster, or cloud provider, including relevant memory and storage.
   - The paper should provide the amount of compute required for each of the individual experimental runs as well as estimate the total compute.
   - The paper should disclose whether the full research project required more compute than the experiments reported in the paper (e.g., preliminary or failed experiments that didn't make it into the paper).

9. **Code of ethics**

   Question: Does the research conducted in the paper conform, in every respect, with the NeurIPS Code of Ethics `https://neurips.cc/public/EthicsGuidelines`?

   Answer: [Yes]

   Justification: We adhere to NeurIPS Code of Ethics.

   Guidelines:

   - The answer [N/A] means that the authors have not reviewed the NeurIPS Code of Ethics.
   - If the authors answer [No], they should explain the special circumstances that require a deviation from the Code of Ethics.
   - The authors should make sure to preserve anonymity (e.g., if there is a special consideration due to laws or regulations in their jurisdiction).

10. **Broader impacts**

    Question: Does the paper discuss both potential positive societal impacts and negative societal impacts of the work performed?

    Answer: [Yes]

    Justification: The paper describes the possibility of reducing training costs and enhancing model generalization performance.

    Guidelines:

- The answer [N/A] means that there is no societal impact of the work performed.
- If the authors answer [N/A] or [No], they should explain why their work has no societal impact or why the paper does not address societal impact.
- Examples of negative societal impacts include potential malicious or unintended uses (e.g., disinformation, generating fake profiles, surveillance), fairness considerations (e.g., deployment of technologies that could make decisions that unfairly impact specific groups), privacy considerations, and security considerations.
- The conference expects that many papers will be foundational research and not tied to particular applications, let alone deployments. However, if there is a direct path to any negative applications, the authors should point it out. For example, it is legitimate to point out that an improvement in the quality of generative models could be used to generate Deepfakes for disinformation. On the other hand, it is not needed to point out that a generic algorithm for optimizing neural networks could enable people to train models that generate Deepfakes faster.
- The authors should consider possible harms that could arise when the technology is being used as intended and functioning correctly, harms that could arise when the technology is being used as intended but gives incorrect results, and harms following from (intentional or unintentional) misuse of the technology.
- If there are negative societal impacts, the authors could also discuss possible mitigation strategies (e.g., gated release of models, providing defenses in addition to attacks, mechanisms for monitoring misuse, mechanisms to monitor how a system learns from feedback over time, improving the efficiency and accessibility of ML).

11. **Safeguards**

    Question: Does the paper describe safeguards that have been put in place for responsible release of data or models that have a high risk for misuse (e.g., pre-trained language models, image generators, or scraped datasets)?

    Answer: [N/A]

    Justification: Our work poses no such risks.

    Guidelines:

    - The answer [N/A] means that the paper poses no such risks.
    - Released models that have a high risk for misuse or dual-use should be released with necessary safeguards to allow for controlled use of the model, for example by requiring that users adhere to usage guidelines or restrictions to access the model or implementing safety filters.
    - Datasets that have been scraped from the Internet could pose safety risks. The authors should describe how they avoided releasing unsafe images.
    - We recognize that providing effective safeguards is challenging, and many papers do not require this, but we encourage authors to take this into account and make a best faith effort.

12. **Licenses for existing assets**

    Question: Are the creators or original owners of assets (e.g., code, data, models), used in the paper, properly credited and are the license and terms of use explicitly mentioned and properly respected?

    Answer: [Yes]

    Justification: The paper cites the original paper that produced the code package or dataset.

    Guidelines:

    - The answer [N/A] means that the paper does not use existing assets.
    - The authors should cite the original paper that produced the code package or dataset.
    - The authors should state which version of the asset is used and, if possible, include a URL.
    - The name of the license (e.g., CC-BY 4.0) should be included for each asset.
    - For scraped data from a particular source (e.g., website), the copyright and terms of service of that source should be provided.

- If assets are released, the license, copyright information, and terms of use in the package should be provided. For popular datasets, `paperswithcode.com/datasets` has curated licenses for some datasets. Their licensing guide can help determine the license of a dataset.
- For existing datasets that are re-packaged, both the original license and the license of the derived asset (if it has changed) should be provided.
- If this information is not available online, the authors are encouraged to reach out to the asset's creators.

13. **New assets**

    Question: Are new assets introduced in the paper well documented and is the documentation provided alongside the assets?

    Answer: [N/A]

    Justification: There are no new assets introduced in this paper.

    Guidelines:

    - The answer [N/A] means that the paper does not release new assets.
    - Researchers should communicate the details of the dataset/code/model as part of their submissions via structured templates. This includes details about training, license, limitations, etc.
    - The paper should discuss whether and how consent was obtained from people whose asset is used.
    - At submission time, remember to anonymize your assets (if applicable). You can either create an anonymized URL or include an anonymized zip file.

14. **Crowdsourcing and research with human subjects**

    Question: For crowdsourcing experiments and research with human subjects, does the paper include the full text of instructions given to participants and screenshots, if applicable, as well as details about compensation (if any)?

    Answer: [N/A]

    Justification: The paper does not involve crowdsourcing nor research with human subjects.

    Guidelines:

    - The answer [N/A] means that the paper does not involve crowdsourcing nor research with human subjects.
    - Including this information in the supplemental material is fine, but if the main contribution of the paper involves human subjects, then as much detail as possible should be included in the main paper.
    - According to the NeurIPS Code of Ethics, workers involved in data collection, curation, or other labor should be paid at least the minimum wage in the country of the data collector.

15. **Institutional review board (IRB) approvals or equivalent for research with human subjects**

    Question: Does the paper describe potential risks incurred by study participants, whether such risks were disclosed to the subjects, and whether Institutional Review Board (IRB) approvals (or an equivalent approval/review based on the requirements of your country or institution) were obtained?

    Answer: [N/A]

    Justification: The paper does not involve crowdsourcing nor research with human subjects.

    Guidelines:

    - The answer [N/A] means that the paper does not involve crowdsourcing nor research with human subjects.
    - Depending on the country in which research is conducted, IRB approval (or equivalent) may be required for any human subjects research. If you obtained IRB approval, you should clearly state this in the paper.

- We recognize that the procedures for this may vary significantly between institutions and locations, and we expect authors to adhere to the NeurIPS Code of Ethics and the guidelines for their institution.
- For initial submissions, do not include any information that would break anonymity (if applicable), such as the institution conducting the review.

16. **Declaration of LLM usage**

    Question: Does the paper describe the usage of LLMs if it is an important, original, or non-standard component of the core methods in this research? Note that if the LLM is used only for writing, editing, or formatting purposes and does *not* impact the core methodology, scientific rigor, or originality of the research, declaration is not required.

    Answer: [N/A]

    Justification: The core method development in this paper does not involve LLMs as any important, original, or non-standard components.

    Guidelines:

    - The answer [N/A] means that the core method development in this research does not involve LLMs as any important, original, or non-standard components.
    - Please refer to our LLM policy in the NeurIPS handbook for what should or should not be described.