Simulating Distribution Dynamics: Liquid Temporal Feature Evolution for Single-Domain Generalized Object Detection

Zihao Zhang[1], Yang Li[1], Aming Wu[2*], Yahong Han[1]

[1]School of Artificial Intelligence, College of Intelligence and Computing, Tianjin University, Tianjin, China
[2]School of Computer Science and Information Engineering, Hefei University of Technology, China
{zhangzihao2490, liyang1398, yahong}@tju.edu.cn, amwu@hfut.edu.cn

Abstract

In this paper, we focus on Single-Domain Generalized Object Detection (Single-DGOD), aiming to transfer a detector trained on one source domain to multiple unknown domains. Existing methods for Single-DGOD typically rely on discrete data augmentation or static perturbation methods to expand data diversity, thereby mitigating the lack of access to target domain data. However, in real-world scenarios such as changes in weather or lighting conditions, domain shifts often occur continuously and gradually. Discrete augmentations and static perturbations fail to effectively capture the dynamic variation of feature distributions, thereby limiting the model's ability to perceive fine-grained cross-domain differences. To this end, we propose a new method, i.e., Liquid Temporal Feature Evolution, which simulates the progressive evolution of features from the source domain to simulated latent distributions by incorporating temporal modeling and liquid neural network–driven parameter adjustment. Specifically, we introduce controllable Gaussian noise injection and multi-scale Gaussian blurring to simulate initial feature perturbations, followed by temporal modeling and a liquid parameter adjustment mechanism to generate adaptive modulation parameters, enabling a smooth and continuous adaptation across domains. By capturing progressive cross-domain feature evolution and dynamically regulating adaptation paths, our method bridges the source-unknown domain distribution gap, significantly boosting generalization and robustness to unseen shifts. Significant performance improvements on the Diverse Weather dataset and Real-to-Art benchmark demonstrate the superiority of our method.

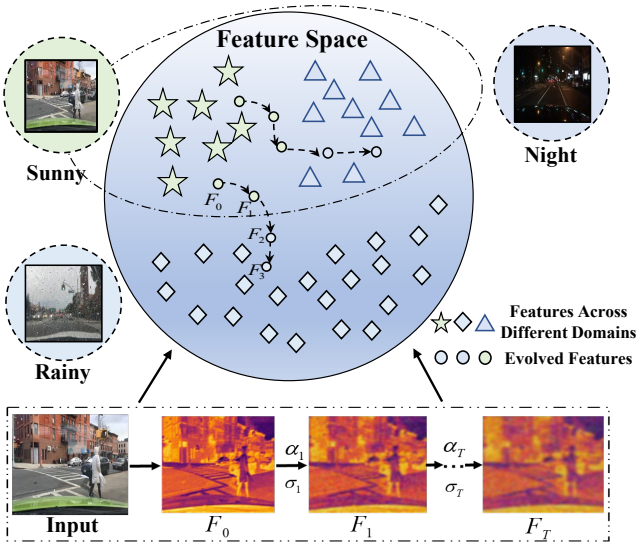

Figure 1: Liquid Temporal Feature Evolution (LTFE) for detecting the unknown-domain object. The core of LTFE lies in simulating the feature evolution trajectory from the source domain to simulated latent distributions, capturing continuous cross-domain feature evolution. As illustrated, LTFE facilitates a smooth transition of features from the source domain to the simulated latent distributions.

Introduction

Single-domain Generalized Object Detection (Vidit 2023; Danish et al. 2024; Liu et al. 2024) is a challenging yet crucial task in object detection, as it requires the model to adapt to domain shifts that were not encountered during training. The core challenges faced in Single-DGOD can be summarized in two main aspects: First, the unobservability of domain shifts (Zhang et al. 2024) prevents models from eliminating the distribution differences between the source domain and unknown domains using traditional feature alignment methods (Wu et al. 2021c,a,b). Second, the diversity of unknown domain data distributions (Danish et al. 2024) makes it difficult for models trained on a single source domain distribution to generalize to multiple unknown domains with different data distributions (Wu and Deng 2023).

Existing Single-DGOD methods primarily enhance generalization to unknown domains through two approaches: one leverages the multimodal capabilities of vision-language models, using static textual prompts from the target domain to estimate domain shifts or simulate target styles (Vidit 2023; Fahes et al. 2023; Li et al. 2024); the other relies on discrete data augmentations to simulate diverse data distributions (Liu et al. 2024; Danish et al. 2024). However, domain shifts in the real world (e.g., weather or lighting changes) typically exhibit continuous and gradual charac-

*Corresponding author.

teristics (Tang et al. 2024; Wu and Deng 2025a), such as transitions from sunny to cloudy to rainy weather. As a result, neither discrete data augmentations nor static textual prompts can capture the continuous evolution of cross-domain features (Wu and Deng 2025b), limiting the model's ability to perceive fine-grained inter-domain differences and, consequently, restricting generalization performance. Moreover, existing textual prompt methods depend on target domain textual priors (Zhang et al. 2025), which conflict with the core assumption of Single-DGOD, further limiting their practical applicability.

To address these challenges, we propose a novel method, i.e., Liquid Temporal Feature Evolution, which simulates the continuous feature evolution trajectory from the source domain to simulated latent distributions (top of Fig. 1) by incorporating temporal modeling and liquid neural network-driven parameter adjustment. This method captures the progressive nature of cross-domain feature evolution, thereby improving generalization to unknown domains. Our method is motivated by two insights: 1) The continuity of domain shifts (Wu, Chen, and Deng 2023; Ganin and Lempitsky 2015) can be modeled through progressive Gaussian perturbations (Lee et al. 2019) to simulate feature evolution; 2) The dynamic adaptation capability of liquid neural networks (Hasani et al. 2021; Kumar et al. 2023) allows adjusting feature evolution paths based on temporal information, facilitating smooth transition between the source domain and simulated distributions. Specifically, we design a collaborative mechanism of multi-scale Gaussian blur and noise to generate continuously evolving features in the feature space (bottom of Fig. 1), simulating the initial feature evolution trajectory from the source domain to simulated distributions. We then model temporal correlations in feature evolution using spatiotemporal memory units, capturing long-range dependencies in the feature sequence. Finally, leveraging liquid neural networks and dynamic evolution information, we construct feature adjustment parameters to achieve smooth feature evolution between the source domain and simulated distributions, enhancing adaptability and robustness to domain shifts. Additionally, to avoid disrupting target features (Wu and Deng 2025b), we constrain the feature evolution process using intra-class consistency and inter-class separability losses, ensuring target features are preserved during evolution from the source domain to simulated distributions.

To evaluate the generalization capability of the proposed method under different types of domain shifts, we conducted experiments on both continuous (Diverse Weather dataset) and discontinuous (Real-to-Art benchmark) domain shift scenarios. Although our method is primarily designed for continuous domain shifts, it also achieves strong performance in discontinuous settings, demonstrating robust generalization. This can be attributed to the rich perturbation space constructed by multi-scale Gaussian blurring and controllable noise injection, as well as the dynamic adaptation of feature evolution paths enabled by the liquid neural network. These components allow the model to flexibly adapt to both gradual and abrupt distribution shifts, exhibiting strong robustness across diverse domain shift patterns.

Related Work

Single-domain Generalization Object Detection

Single-domain generalized object detection (Single-DGOD) is more challenging than multi-domain or domain-adaptive tasks because the model must generalize to multiple unseen target domains using only a single source domain. Although single-domain generalization (SDG) has progressed in image classification (Fan et al. 2021; Qiao, Zhao, and Peng 2020; Wang et al. 2021) and semantic segmentation (Fahes et al. 2023; Jia et al. 2020; Ouyang et al. 2022), advances in object detection remain limited. Existing Single-DGOD methods fall into two categories. Vision-language approaches (Vidit 2023; Fahes et al. 2023) rely on style prompts to infer domain shifts but struggle to capture continuous cross-domain feature variations and conflict with the "unknown target domain" assumption. Static data augmentation methods (e.g., (Danish et al. 2024; Wu et al. 2024)) increase sample diversity through perturbations but fail to model deeper semantic-level domain changes. To overcome these limitations, we propose leveraging simulated feature evolution trajectories to better guide generalization toward unseen target domains.

Liquid Neural Network

Liquid Neural Networks (Hasani et al. 2021) have gained attention for their ability to dynamically adapt to changing environments and process temporal information. Unlike traditional fixed-architecture networks, they simulate the continuous evolution of information through dynamic recurrent connections, allowing them to handle complex, time-varying data. Their key advantage is enhanced robustness and generalization when facing input noise, environmental changes, and data uncertainty. While primarily applied in robotic control and time-series forecasting (Kumar et al. 2023; Hasani et al. 2022), their use in object detection and domain generalization remains limited. In this paper, we leverage the dynamic adaptability and generalization of Liquid Neural Networks to dynamically adjust the evolving feature trajectory, enabling a smooth and continuous transition between the source domain and latent domain distribution.

Method

As shown in Figure 2, to address the Single-Domain Generalized Object Detection (Single-DGOD) problem, we propose a Liquid Temporal Feature Evolution (LTFE) method. By simulating the evolutionary trajectory of feature distributions from the source domain to the simulated latent distributions, it captures the continuous variation patterns of features across domains, thereby enhancing the model's generalization ability in the unknown domains. The overall training and inference process is shown in Algorithm 1.

Progressive Temporal Feature Evolution

To simulate the progressive evolution of features from a source domain to a latent distribution, we introduce a sequential perturbation mechanism that models the underlying feature shift. We follow the baseline work and exploit the widely used object detector, i.e., Faster R-CNN (Ren et al.

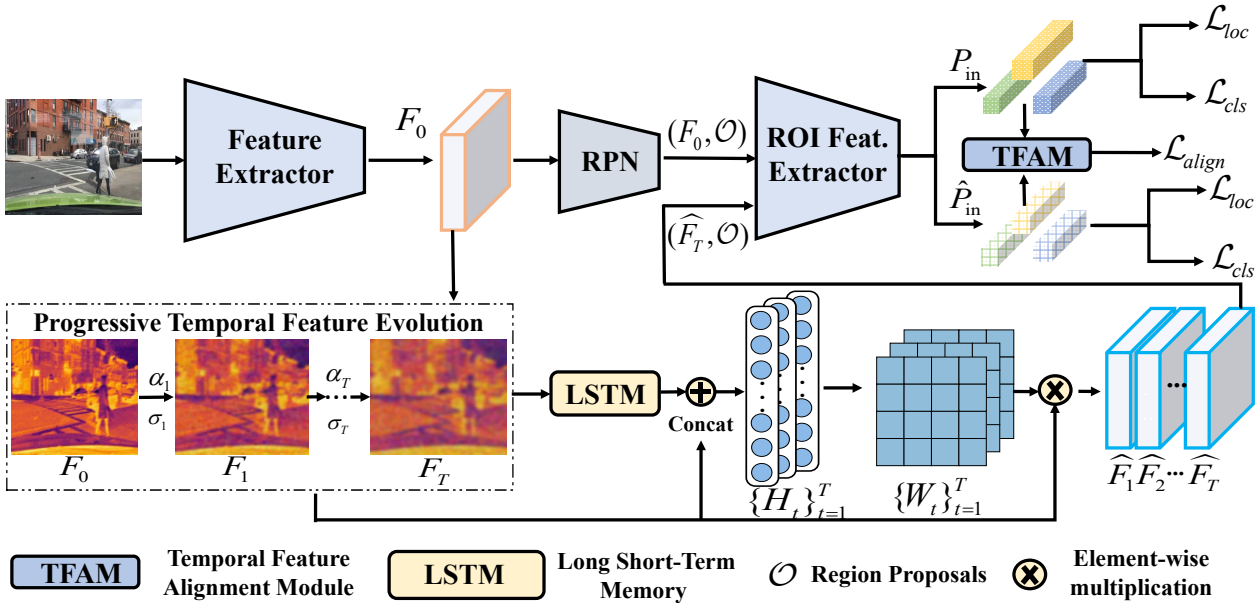

Figure 2: Illustration of Liquid Temporal Feature Evolution. First, the first-layer feature map F_0 extracted by the backbone network is iteratively perturbed to generate the initial feature sequence $\{F_t\}_{t=1}^T$, simulating the feature transitions from the source domain to a latent domain. Next, LSTM is employed to model the generated feature sequence, capturing the spatiotemporal dependencies of feature evolution. Then, liquid parameter evolution is applied to dynamically adjust the original feature sequence, yielding the final evolved sequence $\{\hat{F}_t\}_{t=1}^T$, enabling a smooth and continuous transition between the source domain and the potential domain distributions. Finally, an alignment loss is used to further constrain the entire feature evolution process.

2015), as the basic detection model. Given an input image, the backbone network (e.g. ResNet) extracts an initial feature map $F_0 \in \mathbb{R}^{w \times h \times c}$, where w, h, and c denote the width, height, and channel dimension, respectively. Experimental results show that evolving features in the first layer and using them to generate layers 2–4 leads to the best performance, as detailed in the supplementary material.

To emulate cross-domain transition, we progressively perturb the features. At each time step t, the feature map is transformed by applying a combination of Gaussian blurring (Lee et al. 2019) and stochastic noise injection:

$$F_t = G(\sigma_t) * F_{t-1} + \alpha_t \cdot G(\sigma_t), \tag{1}$$

where σ_t controls the degree of blurring, α_t determines the noise intensity, and $G(\sigma_t)$ denotes the Gaussian blur kernel with variance σ_t, defined as:

$$G(\sigma_t) = \frac{1}{2\pi\sigma_t^2} \exp\left(-\frac{i^2 + j^2}{2\sigma_t^2}\right), \tag{2}$$

where (i, j) represents the spatial coordinates in the kernel. The first term progressively blurs the image to simulate visual degradation, while the second injects Gaussian noise with exponentially decaying intensity:

$$\alpha_t = \alpha_0 \cdot \exp(-\lambda t), \tag{3}$$

where α_t follows an exponential decay to ensure that noise influence diminishes over time, aligning with realistic domain shift patterns. And λ is a hyperparameter set to 0.2, controlling the rate at which the noise intensity decays over time. The baseline noise intensity α_0 is set to 0.2, while the blurring intensity increases exponentially:

$$\sigma_t = \sigma_0 \cdot \gamma^t, \tag{4}$$

where γ is a hyperparameter set to 1.2, controlling the degradation rate, ensuring that features transition progressively from a fine-grained state ($\sigma_0 = 1$) to a coarser representation (σ_t increasing).

Temporal Dependency Modeling

To capture temporal dependency patterns in feature evolution trajectories, we feed the simulated feature sequence $\{F_t\}_{t=1}^T$ generated by temporal perturbations into an LSTM network (Yu et al. 2019) for spatiotemporal interaction modeling. The memory state propagation is implemented through the following gated mechanism:

$$h_t, c_t = \text{LSTM}(F_t, h_{t-1}, c_{t-1}), \tag{5}$$

where $h_t \in \mathbb{R}^d$ and $c_t \in \mathbb{R}^d$ denote the hidden state and cell state at timestep t, respectively. The hidden state h_t captures feature abstractions at the current time step, while the cell state c_t maintains long-term memory. To enhance the interaction between original features and historical memory, we introduce a feature-state fusion mechanism:

$$H_t = \text{ReLU}(W_p[h_t \oplus F_t]), \tag{6}$$

where $W_p \in \mathbb{R}^{(d+c) \times d}$ represents a learnable projection matrix, and $\oplus$ indicates channel-wise concatenation. The final temporal encoding features $H = \{H_t\}_{t=1}^T \in \mathbb{R}^{T \times d}$, where

Algorithm 1: Liquid Temporal Feature Evolution

Input: Input image I, initial feature map F_0, hyperparameters $\alpha_0, \lambda, \gamma$, number of steps T
Output: Evolved feature sequence $\{\hat{F}_t\}_{t=1}^{T}$

Training Phase: while *train* **do**
  Initialize F_0;
  for $t = 1$ **to** T **do**
    $\sigma_t = \sigma_0 \cdot \gamma^t$, $\alpha_t = \alpha_0 \cdot \exp(-\lambda t)$;
    $F_t = G(\sigma_t) * F_{t-1} + \alpha_t \cdot G(\sigma_t)$;
  Feed $\{F_t\}_{t=1}^{T}$ into LSTM and obtain $\{H_t\}_{t=1}^{T}$;
  for $t = 1$ **to** T **do**
    $H_t = \text{ReLU}(W_p[h_t \oplus F_t])$;
  Solve ODE: $W_t = \text{ODESolve}(f_\theta, W_0, H_t, t)$;
  Adjust the feature: $\hat{F}_t = \text{Conv2D}(F_0, W_t) + F_0$;

Inference Phase: while *eval* **do**
  Extract feature F_0 from input image I;
  Simulate temporal feature perturbations for 1-2 steps to generate $\{F_t\}_{t=1}^{2}$;
  Generate dynamic convolution kernel $W_{\text{test}} = \text{ODESolve}(f_\theta, W_0, H_t, \tau)$;
  Adjust feature map: $\hat{F}_t = \text{Conv2D}(F_0, W_{\text{test}}) + F_0$;
  Classify and localize using $\hat{F}_T$;

d is the hidden dimension, comprehensively capture the evolution patterns from the initial state F_1 to the final state F_T. This feature sequence offers spatio-temporally consistent representations for generating dynamic parameters.

Liquid Parameter Evolution

To ensure a continuous and smooth transition between the source and latent distribution, we construct a dynamic convolution kernel generation network based on Neural Ordinary Differential Equations (Chen et al. 2018) to adjust the evolved features, as described in the following equation:

$$\frac{dW(\tau)}{d\tau} = f_\theta\left(W(\tau), H_t\right), \tag{7}$$

where $f_\theta : \mathbb{R}^d \to \mathbb{R}^{k\times k\times c_{\text{in}}\times c_{\text{out}}}$ is a vector field function implemented using a liquid neural network, $H_t \in \mathbb{R}^d$ denotes the temporal feature encoding at time step t, and $W(\tau)$ is the convolution kernel parameter evolving along a continuous trajectory governed by virtual time τ. The kernel size is $k \times k$, and c_{in}, c_{out} are the input and output channels.

The fourth-order Runge-Kutta (Hult 2007) method is used to numerically solve the ODE (Chen et al. 2018), generating the time-varying convolution kernel. The process is described as follows:

$$W_t = \text{ODESolve}\left(f_\theta, W_0, H_t, \tau\right), \tag{8}$$

where $W_0 \in \mathbb{R}^{k\times k\times c_{\text{in}}\times c_{\text{out}}}$ is the initial kernel (inherited from the pre-trained Faster R-CNN), and the integration interval $\tau \in [0, \hat{\tau}]$ is dynamically determined by the L2 norm of the current temporal feature encoding H_t:

$$\hat{\tau} = \frac{\|H_t\|_2}{\max \|H_t\|_2}, \tag{9}$$

which normalizes evolution time by the sequence's maximum feature magnitude, ensuring larger temporal deviations (higher $\|H_t\|_2$) result in longer kernel evolution and improved adaptation to significant domain shifts. The dynamically generated kernel is then applied to the initial feature F_0 for input-conditioned adjustment:

$$\hat{F}_t = \text{Conv2D}\left(F_0, W_t\right) + F_0, \tag{10}$$

where the residual connection preserves original semantic information. With a fixed $k \times k$ ($k = 3$) convolution kernel W_t and consistent channels, a dynamic feature sequence $\{\hat{F}_t\}_{t=1}^{T}$ is produced, providing continuous training signals. Though only $\hat{F}_T$ is used in inference, LSTM-based temporal modeling of intermediate features enables learning of domain shift dynamics, ensuring $\hat{F}_T$ reflects the full evolution trajectory and improves generalization to unseen domains.

Temporal Feature Alignment Module

To ensure semantic consistency of target features during feature evolution and mitigate noise interference, we introduce feature alignment using intra-class consistency and inter-class separability losses. Specifically, the original feature F_0 is fed into the RPN to generate proposals $\mathcal{O}$. Then, both F_0 and the final evolved feature $\hat{F}_T$, along with $\mathcal{O}$, are passed through the ROI head to extract proposal-level features $P_{\text{in}} \in \mathbb{R}^{m\times n}$ and $\hat{P}_{\text{in}} \in \mathbb{R}^{m\times n}$, where m and n denote the number of proposals and feature dimensions. Notably, the intermediate feature sequence models continuous domain shifts via progressive perturbation and temporal dynamics, while the alignment loss enforces consistency in target semantics throughout the evolution.

Next, we minimize the feature distance of same-class instances across domain evolution using P_{in} and $\hat{P}_{\text{in}}$:

$$\mathcal{L}_{\text{intra}} = \frac{1}{N}\sum_{i=1}^{N} \|P_{\text{in}}^{(i)} - \hat{P}_{\text{in}}^{(i)}\|_2^2. \tag{11}$$

At the same time, we maximize the feature differences between instances of different classes as follows:

$$\mathcal{L}_{\text{inter}} = -\log \frac{\exp(s(P_{\text{in}}^{(i)}, \hat{P}_{\text{in}}^{(i)}))}{\sum_{j\neq i}\exp(s(P_{\text{in}}^{(i)}, \hat{P}_{\text{in}}^{(j)}))}, \tag{12}$$

where $s(\cdot)$ is the cosine similarity function. The total alignment loss is defined as:

$$\mathcal{L}_{\text{align}} = \lambda_1\mathcal{L}_{\text{intra}} + \lambda_2\mathcal{L}_{\text{inter}}, \tag{13}$$

which λ_1 and λ_2 are weight balancing coefficients, with $\lambda_1 = 1.0$ and $\lambda_2 = 0.1$ as fixed values. Then, we input the example features P_{in} and $\hat{P}_{\text{in}}$ into the object classifier and regressor to calculate the classification loss $\mathcal{L}_{\text{cls}}$ and localization loss $\mathcal{L}_{\text{loc}}$. The joint objective is defined as:

$$\mathcal{L}_{\text{total}} = \mathcal{L}_{\text{cls}} + \mathcal{L}_{\text{reg}} + \mathcal{L}_{\text{align}}, \tag{14}$$

where $\mathcal{L}_{\text{cls}}$ and $\mathcal{L}_{\text{reg}}$ are the classification and regression losses, respectively.

Inference for Target Domain Object Detection

During inference, unlike training, the model requires only minimal feature evolution, performing limited perturbations to fine-tune adaptation to the real target domain and bridge the gap between simulated training and true distributions. Typically, two discrete iterations ($T = 2$) suffice for adaptive feature adjustment. The process begins with image feature extraction to generate the initial evolved feature sequence $\{F_t\}_{t=1}^{2}$, which is then refined via the Temporal Dependency Modeling and Liquid Parameter Evolution. In this mechanism, the Liquid Neural Network (LNN) employs ODE-based parameter evolution to dynamically generate convolution kernels, not as random augmentation, but as a controlled adjustment to maintain the semantic consistency of target features during perturbation. The final adjusted feature $\hat{F}_2$ is used for classification and localization. This approach enables targeted feature evolution and dynamic adaptation during inference, allowing the model to align with the target domain's feature distribution while preserving critical semantic information. Algorithm 1 details the training and inference procedures.

Experiments

In the experiments, for Single-DGOD, we follow the settings of the work (Wu and Deng 2022; Li et al. 2024; Liu et al. 2024) to validate the model's generalization capability. Furthermore, to validate the effectiveness of our method, we evaluated the model on the Reality-to-Art generalization benchmark (Danish et al. 2024).

Experimental setup

Dataset. The entire experiment is conducted based on the Diverse-Weather and Real-to-Art benchmarks. **Diverse Driving Weather Scenarios.** To ensure a fair comparison, we use the same training and testing datasets as other Single-DGOD methods (Wu and Deng 2022; Liu et al. 2024; Danish et al. 2024). Training is conducted only on the daytime sunny dataset, followed by direct testing on four target domains with different weather conditions: nighttime clear, dusk rainy, nighttime rainy, and daytime foggy. The category space remains consistent across all datasets. **Generalization from Reality to Art.** Following (Inoue et al. 2018; Danish et al. 2024), we train on the Pascal VOC2007 and VOC2012 trainval sets (Everingham et al. 2010), and evaluate generalization on Clipart1k, Watercolor2k, and Comic2k. Clipart shares all 20 Pascal VOC classes, while Watercolor2k and Comic2k include 6-class subsets.

Metric. We adopt the same evaluation metrics as Single-DGOD (Wu and Deng 2022) to ensure a fair comparison with other methods. We use Mean Average Precision (mAP) with an IoU threshold of 0.5 to evaluate the model's performance across various datasets and categories.

Implementation Details

For a fair comparison with other Single-DGOD methods (Wu and Deng 2022; Liu et al. 2024), we use the same baseline model, Faster R-CNN (Ren et al. 2015). While prior works report results only with ResNet-101 (He et al. 2016), we provide a more comprehensive evaluation using ResNet-50, ResNet-101, and Swin Transformer (Liu et al. 2021). The model achieves optimal performance during training when $T = 8$ in the Progressive Temporal Feature Evolution process. In Equation (1), α_0 and σ_0 are initialized to 0.2 and 1, respectively. For inference, T is set to 2 to preserve target domain information. Training is conducted with SGD (momentum = 0.9) for 20 epochs, starting with a learning rate of 0.02. All experiments run on two NVIDIA RTX 3090 GPUs (24GB) with images resized to 800×800 pixels.

Method	mAP				
	Day Clear	Night Sunny	Dusk Rainy	Night Rainy	Day Foggy
Faster R-CNN(Res101) (Ren et al. 2015)	48.1	34.4	26.0	12.4	32.0
SW(Res101) (Pan et al. 2019)	50.6	33.4	26.3	13.7	30.8
IBN-Net(Res101) (Pan et al. 2018)	49.7	32.1	26.1	14.3	29.6
IterNorm(Res101) (Huang et al. 2019)	43.9	29.6	22.8	12.6	28.4
ISW(Res101) (Choi et al. 2021)	51.3	33.2	25.9	14.1	31.8
S-DGOD (Res101)(Wu and Deng 2022)	56.1	36.6	28.2	16.6	33.5
C-Gap (Res101)(Vidit 2023)	51.3	36.9	32.3	18.7	38.5
PDOC (Res101)(Li et al. 2024)	53.6	38.5	33.7	19.2	39.1
UFR (Res101)(Liu et al. 2024)	58.6	40.8	33.2	19.2	39.6
DIV (Res101) (Danish et al. 2024)	52.8	42.5	38.1	24.1	37.2
G-NAS (Res101) (Wu et al. 2024)	58.4	45.0	35.1	17.4	36.4
SECOT (Res101) (Zhang et al. 2025)	55.4	42.0	39.2	24.5	40.6
FWCL (Res101) (Guo et al. 2025)	55.5	37.5	32.6	18.9	32.3
Ours(Res50)	50.9	31.2	29.4	16.7	35.4
Ours(Res101)	58.4	43.1	39.7	24.3	41.2
Ours(Swin-T)	**64.2**	**53.1**	**46.5**	**33.4**	**46.4**

Table 1: Single-Domain Generalization Results (mAP(%)) in Diverse Weather Driving Scenarios.

Comparison with the State of the Art

Diverse Driving Weather Scenarios. We compare our method with state-of-the-art Single-DGOD methods (Liu et al. 2024; Wu et al. 2024), as shown in Table 1. Under scenarios with continuous domain bias, our method achieves superior performance across all four target domains. Figure 3 presents detection results under four weather conditions, demonstrating that our model maintains strong performance even in challenging scenarios such as rainy nights.

Method	mAP			
	VOC	Comic	Watercolor	Clipart
Faster R-CNN(Res101)(Ren et al. 2015)	80.4	19.4	45.6	26.5
NP(Res101)(Fan et al. 2023)	79.2	28.9	53.3	35.4
C-Gap(Res101)(Vidit 2023)	80.5	29.4	50.7	36.7
DIV(Res101)(Danish et al. 2024)	80.1	33.2	57.4	38.9
SECOT(Res101)(Zhang et al. 2025)	82.9	34.8	57.5	40.2
Ours(Res50)	80.2	27.4	54.1	35.6
Ours(Res101)	84.6	35.4	58.2	42.1
Ours(Swin-T)	**88.2**	**37.3**	**61.3**	**44.5**

Table 2: Single-Domain Generalization Results (mAP(%)) from Real to Artistic. **Bold values** indicate the best results.

Generalization from Reality to Art. We evaluate the generalization ability of our method across real and artistic domains, a representative case of non-continuous domain bias with large distribution gaps. As shown in Table 2, using the same ResNet-101 backbone, our model consistently outperforms others on three artistic-style datasets. For instance, compared to the SECOT(Zhang et al. 2025), it achieves a 4.7% improvement on the Cliparts dataset.

Comparison with Continual Test-Time Adaptation Methods. Since we introduced parameter adjustment during

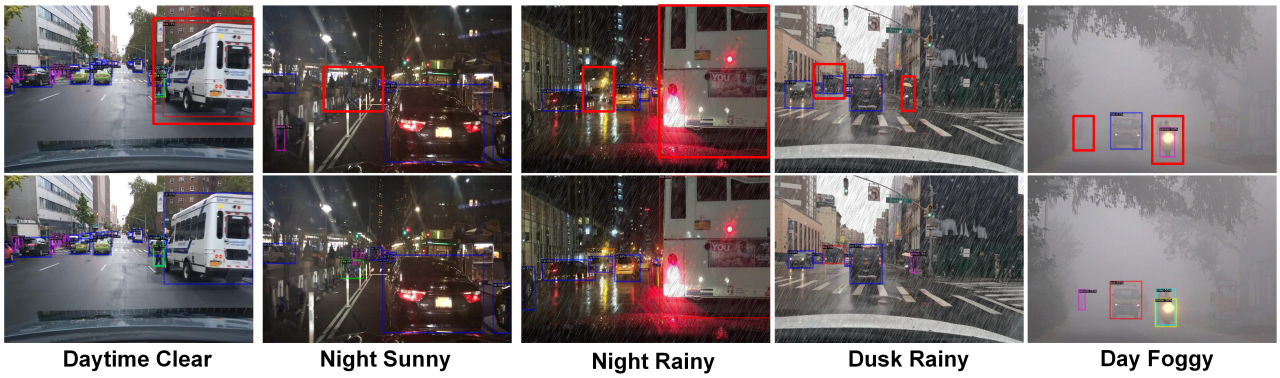

Figure 3: Qualitative Results: Detection results under different weather conditions. The first and second rows display the results from G-NAS (Wu et al. 2024) and our method, respectively. To provide a more intuitive comparison, we highlight the objects missed or incorrectly detected by G-NAS (Wu et al. 2024) using red boxes, which are correctly identified by our method.

Method	mAP			
	Night Sunny	Dusk Rainy	Night Rainy	Day Foggy
Baseline	35.4	27.5	13.7	32.6
MemCLR (VS et al. 2023)	33.5	32.7	16.4	36.7
SKIP (Yoo et al. 2024)	32.6	34.1	17.3	32.9
ECPG (Li et al. 2025)	35.3	35.7	18.2	33.7
Direct-Test	42.4	37.9	23.6	38.9
Ours	**43.1**	**39.7**	**24.3**	**41.2**

Table 3: Continual test-time adaptive object detection results (%) on the Diverse-Weather dataset using the ResNet-101.

feature evolution to control the trajectory from the source domain to the target domain, we can adapt to the target domain distribution by fine-tuning parameters during the test phase. Therefore, we compared our method with the latest continual test-time adaptation methods (Yoo et al. 2024; Li et al. 2025), as shown in Table 3. Our method outperforms test-time adaptation across all four target domains, notably improving night-sunny from 35.3% to 43.1% with ECPG, and demonstrates that training-time feature evolution enhances target-domain adaptation and generalization.

Model	**mAP (%)**	**Params (M)**	**FPS**	**FLOPs (G)**
Faster R-CNN (Ren et al. 2015)	26.2	90.8	4.7	267
S-DGOD (Wu and Deng 2022)	28.7	120.4	3.9	284
C-Gap (Vidit 2023)	31.6	190.2	2.5	423
PDOC (Li et al. 2024)	32.6	198.6	2.2	478
UFR (Liu et al. 2024)	33.2	184.2	6.9	323
Ours(Res101)	37.1	140.6	6.4	325

Table 4: Model Comparison in Terms of mAP, Parameters, FPS, and FLOPs.

Computational Efficiency and Time Complexity Analysis. We compare the time complexity and computational efficiency of existing methods in Table 4. As can be seen, compared to other single-domain generalization methods, such as C-CAP(Vidit 2023) and PDOC(Li et al. 2024), our approach achieves superior performance with lower computational cost and reduced latency, demonstrating its efficiency and deployability.

Method			Source	Target		
PTFE	TDM+LPE	TFAM	Day Clear	Night Sunny	Dusk Rainy	Night Rainy
			49.2	35.4	27.5	13.7
✓			54.6	36.8	29.3	14.1
✓	✓		57.3	42.6	39.4	22.7
✓		✓	56.8	40.2	34.1	19.7
✓	✓	✓	**58.4**	**43.1**	**39.7**	**24.3**

Table 5: Ablation analysis of our proposed model.

Ablation Study

Component Analysis. To validate the effectiveness of each component, we conducted ablation studies on Progressive Temporal Feature Evolution (PTFE), the combined Temporal Dependency Modeling (TDM) and Liquid Parameter Evolution (LPE) module, and Temporal Feature Alignment Module (TFAM). Since LPE relies on temporal information for dynamic adjustment, it is evaluated jointly with TDM. Results are shown in Table 5, with the first row representing the Faster R-CNN baseline. The second row adds PTFE to simulate the initial feature evolution without dynamic optimization. The third row further incorporates TDM and LPE, effectively capturing cross-domain variations and increasing data diversity during training, leading to significant performance gains over the second row. The fourth row adds TFAM to PTFE to reduce target information loss during evolution. The final row shows the full model, where dynamic control of the feature evolution trajectory better approximates the distribution shift from source to potential target domains, significantly enhancing generalization.

The Impact of Time Step T in the Evolution of Progressive Temporal Feature Evolution. The time step T is a key hyperparameter in Progressive Temporal Feature Evolution (PTFE), determining both the length and granularity of the feature evolution process. A larger T during training extends the evolution trajectory, enabling a more thorough simulation of domain shifts and improving cross-domain adaptability. As shown in Figure 5, increasing T steadily boosts mAP(%) on the 'Day Foggy' and 'Dusk Rainy' domains, converging around $T = 8$. However, during inference, an overly large T may distort semantic content and accumu-

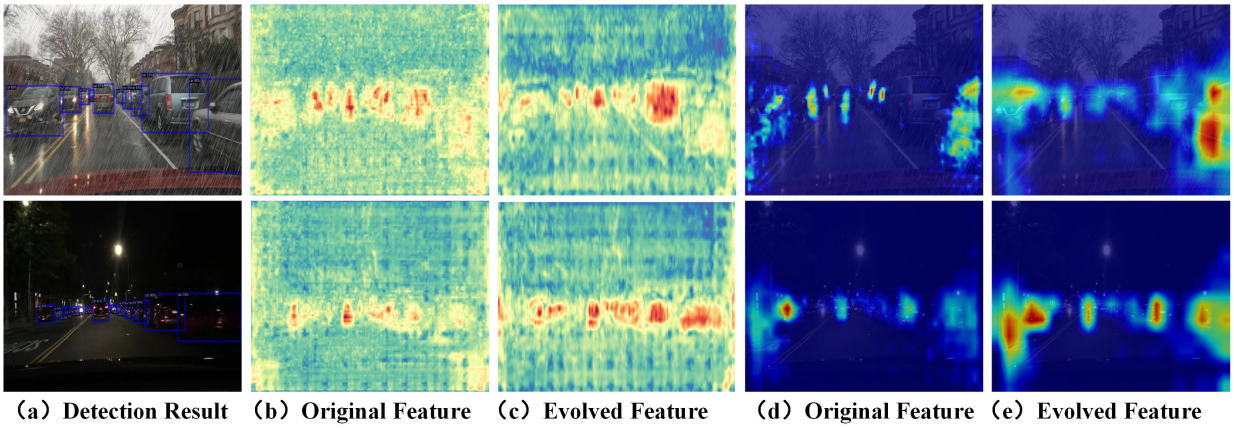

Figure 4: Visualization analysis of our method: The first column displays the model's detection results. The second and third columns correspond to the channel activation visualizations of the initial feature F_0 and the evolved feature $\hat{F}_T$, respectively. The fourth and fifth columns display the heatmap visualizations of the initial feature F_0 and the evolved feature $\hat{F}_T$.

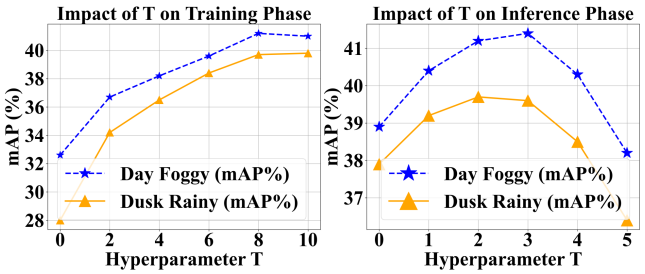

Figure 5: Analysis of the time step T in the Progressive Temporal Feature Evolution.

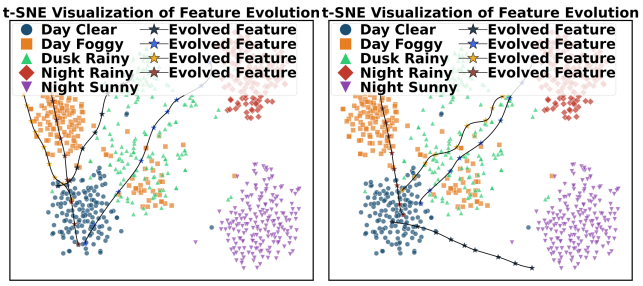

Figure 6: t-SNE Visualization Analysis of Liquid Temporal Feature Evolution.

late noise, reducing accuracy. As illustrated in Figure 5, the best inference performance is achieved when $T = 2$.

t-SNE Visualization of Liquid Temporal Feature Evolution. To validate the effectiveness of the progressively shifted feature evolution in our proposed Liquid Temporal Feature Evolution (LTFE), we conducted t-SNE visualization on the generated evolving feature sequence, as shown in Fig. 6. By mapping features from different time steps into a two-dimensional space, we can visually observe the evolution of features between the source and target domains. The visualization results show that as the time steps increase, features from the source domain progressively shift toward those of the target domain. This confirms that our method effectively simulates the data distribution of the potential target domain through gradual perturbations, validating the effectiveness of the proposed LTFE.

Visualization Analysis

In Figure 4, we present channel activation and heatmap visualizations to compare the initial feature F_0 with the final evolved feature $\hat{F}_T$ during the testing phase. The results show that the final evolved feature $\hat{F}_T$ exhibits stronger activation values across multiple channels, indicating that the model captures key features of the target domain more effectively during feature evolution. This enhanced activation highlights the model's attention to crucial information in the target domain, particularly in complex backgrounds or across different object categories. Furthermore, heatmap visualization shows that the final evolved feature $\hat{F}_T$ focuses more on foreground objects, indicating improved localization and recognition in detection tasks. These results confirm that simulating latent distribution shifts progressively refines features and improves generalization to unseen domains.

Conclusion

In the Single-DGOD task, we propose a novel method, Liquid Temporal Feature Evolution, which enhances the model's generalization capability in potential target domains by simulating the feature evolution trajectory from the source domain to unseen target domains. Our method combines Progressive Temporal Feature Evolution, Temporal Dependency Modeling, and Liquid Parameter Evolution to smoothly transition from the source domain to latent distributions, while intra-class consistency and inter-class separability losses preserve feature integrity during evolution. Significant gains on the Diverse Weather dataset and Real-to-Art benchmark confirm its effectiveness.

Acknowledgments

This work is supported by the National Nature Science Foundation of China (Nos. 62376186, 62472333).

References

Chen, R. T.; Rubanova, Y.; Bettencourt, J.; and Duvenaud, D. K. 2018. Neural ordinary differential equations. *Advances in neural information processing systems*, 31.

Choi, S.; Jung, S.; Yun, H.; Kim, J. T.; Kim, S.; and Choo, J. 2021. RobustNet: Improving Domain Generalization in Urban-Scene Segmentation via Instance Selective Whitening. In *2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*.

Danish, M. S.; Khan, M. H.; Munir, M. A.; Sarfraz, M. S.; and Ali, M. 2024. 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*, 17732–17742.

Everingham, M.; Van Gool, L.; Williams, C. K.; Winn, J.; and Zisserman, A. 2010. The pascal visual object classes (voc) challenge. *International journal of computer vision*, 88: 303–338.

Fahes, M.; Vu, T.-H.; Bursuc, A.; Pérez, P.; and de Charette, R. 2023. PODA: Prompt-driven Zero-shot Domain Adaptation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 18623–18633.

Fan, Q.; Segu, M.; Tai, Y.-W.; Yu, F.; Tang, C.-K.; Schiele, B.; and Dai, D. 2023. Towards robust object detection invariant to real-world domain shifts. In *The Eleventh International Conference on Learning Representations (ICLR 2023)*. OpenReview.

Fan, X.; Wang, Q.; Ke, J.; Yang, F.; Gong, B.; and Zhou, M. 2021. Adversarially adaptive normalization for single domain generalization. In *Proceedings of the IEEE/CVF conference on Computer Vision and Pattern Recognition*, 8208–8217.

Ganin, Y.; and Lempitsky, V. 2015. Unsupervised domain adaptation by backpropagation. In *International conference on machine learning*, 1180–1189. PMLR.

Guo, X.; Liu, C.; Qian, X.; Wang, Z.; Feng, X.; and Xue, Y. 2025. Single-Domain Generalized Object Detection with Frequency Whitening and Contrastive Learning. *IEEE Transactions on Multimedia*, 1–14.

Hasani, R.; Lechner, M.; Amini, A.; Liebenwein, L.; Ray, A.; Tschaikowski, M.; Teschl, G.; and Rus, D. 2022. Closed-form continuous-time neural networks. *Nature Machine Intelligence*, 4(11): 992–1003.

Hasani, R.; Lechner, M.; Amini, A.; Rus, D.; and Grosu, R. 2021. Liquid time-constant networks. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 35, 7657–7666.

He, K.; Zhang, X.; Ren, S.; and Sun, J. 2016. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, 770–778.

Huang, L.; Zhou, Y.; Zhu, F.; Liu, L.; and Shao, L. 2019. Iterative Normalization: Beyond Standardization towards Efficient Whitening. *Cornell University - arXiv,Cornell University - arXiv*.

Hult, J. 2007. A fourth-order Runge–Kutta in the interaction picture method for simulating supercontinuum generation in optical fibers. *Journal of lightwave technology*, 25(12): 3770–3775.

Inoue, N.; Furuta, R.; Yamasaki, T.; and Aizawa, K. 2018. Cross-domain weakly-supervised object detection through progressive domain adaptation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, 5001–5009.

Jia, Y.; Zhang, J.; Shan, S.; and Chen, X. 2020. Single-side domain generalization for face anti-spoofing. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 8484–8493.

Kumar, K.; Verma, A.; Gupta, N.; and Yadav, A. 2023. Liquid Neural Networks: A Novel Approach to Dynamic Information Processing. In *2023 International Conference on Advances in Computation, Communication and Information Technology (ICAICCIT)*, 725–730.

Lee, J.; Lee, S.; Cho, S.; and Lee, S. 2019. Deep defocus map estimation using domain adaptation. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 12222–12230.

Li, D.; Wu, A.; Li, Y.; Wang, Y.; and Han, Y. 2025. Continual Adaptation: Environment-Conditional Parameter Generation for Object Detection in Dynamic Scenarios. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*.

Li, D.; Wu, A.; Wang, Y.; and Han, Y. 2024. Prompt-Driven Dynamic Object-Centric Learning for Single Domain Generalization. *arXiv preprint arXiv:2402.18447*.

Liu, Y.; Zhou, S.; Liu, X.; Hao, C.; Fan, B.; and Tian, J. 2024. Unbiased Faster R-CNN for Single-source Domain Generalized Object Detection. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 28838–28847.

Liu, Z.; Lin, Y.; Cao, Y.; Hu, H.; Wei, Y.; Zhang, Z.; Lin, S.; and Guo, B. 2021. Swin transformer: Hierarchical vision transformer using shifted windows. In *Proceedings of the IEEE/CVF international conference on computer vision*, 10012–10022.

Ouyang, C.; Chen, C.; Li, S.; Li, Z.; Qin, C.; Bai, W.; and Rueckert, D. 2022. Causality-inspired single-source domain generalization for medical image segmentation. *IEEE Transactions on Medical Imaging*, 42(4): 1095–1106.

Pan, X.; Luo, P.; Shi, J.; and Tang, X. 2018. *Two at Once: Enhancing Learning and Generalization Capacities via IBN-Net*, 484–500.

Pan, X.; Zhan, X.; Shi, J.; Tang, X.; and Luo, P. 2019. Switchable Whitening for Deep Representation Learning. In *2019 IEEE/CVF International Conference on Computer Vision (ICCV)*.

Qiao, F.; Zhao, L.; and Peng, X. 2020. Learning to Learn Single Domain Generalization. In *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*.

Ren, S.; He, K.; Girshick, R.; and Sun, J. 2015. Faster r-cnn: Towards real-time object detection with region proposal networks. *Advances in neural information processing systems*, 28.

Tang, S.; Su, W.; Ye, M.; and Zhu, X. 2024. Source-free domain adaptation with frozen multimodal foundation model. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 23711–23720.

Vidit. 2023. Clip the gap: A single domain generalization approach for object detection. 3219–3229.

VS, V.; Oza, P.; Patel, V. M.; and Patel, V. M. 2023. Towards Online Domain Adaptive Object Detection. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, 478–488.

Wang, Z.; Luo, Y.; Qiu, R.; Huang, Z.; and Baktashmotlagh, M. 2021. Learning To Diversify for Single Domain Generalization. *International Conference on Computer Vision,International Conference on Computer Vision*.

Wu, A.; Chen, D.; and Deng, C. 2023. Deep feature deblurring diffusion for detecting out-of-distribution objects. In *Proceedings of the IEEE/CVF international conference on computer vision*, 13381–13391.

Wu, A.; and Deng, C. 2022. 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*, 847–856.

Wu, A.; and Deng, C. 2023. Discriminating Known From Unknown Objects via Structure-Enhanced Recurrent Variational AutoEncoder. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 23956–23965.

Wu, A.; and Deng, C. 2025a. Percept, Memory, and Imagine: World Feature Simulating for Open-Domain Unknown Object Detection. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, 4682–4691.

Wu, A.; and Deng, C. 2025b. Towards OOD Object Detection with Unknown-Concept Guided Feature Diffusion. *IEEE Transactions on Pattern Analysis and Machine Intelligence*.

Wu, A.; Han, Y.; Zhu, L.; and Yang, Y. 2021a. Instance-invariant domain adaptive object detection via progressive disentanglement. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 44(8): 4178–4193.

Wu, A.; Han, Y.; Zhu, L.; and Yang, Y. 2021b. Universal-prototype enhancing for few-shot object detection. In *Proceedings of the IEEE/CVF international conference on computer vision*, 9567–9576.

Wu, A.; Liu, R.; Han, Y.; Zhu, L.; and Yang, Y. 2021c. Vector-decomposed disentanglement for domain-invariant object detection. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 9342–9351.

Wu, F.; Gao, J.; Hong, L.; Wang, X.; Zhou, C.; and Ye, N. 2024. G-NAS: Generalizable Neural Architecture Search for Single Domain Generalization Object Detection. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 38, 5958–5966.

Yoo, J.; Lee, D.; Chung, I.; Kim, D.; and Kwak, N. 2024. What, How, and When Should Object Detectors Update in Continually Changing Test Domains? In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 23354–23363.

Yu, Y.; Si, X.; Hu, C.; and Zhang, J. 2019. A review of recurrent neural networks: LSTM cells and network architectures. *Neural computation*, 31(7): 1235–1270.

Zhang, J.; Yang, K.; Shi, H.; Reiß, S.; Peng, K.; Ma, C.; Fu, H.; Torr, P. H.; Wang, K.; and Stiefelhagen, R. 2024. Behind every domain there is a shift: Adapting distortion-aware vision transformers for panoramic semantic segmentation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*.

Zhang, Z.; Wu, A.; Han, Y.; Zhang, Z.; Wu, A.; and Han, Y. 2025. Style Evolving along Chain-of-Thought for Unknown-Domain Object Detection. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, 14225–14234.