MaskViM: Domain Generalized Semantic Segmentation with State Space Models

Jiahao Li[1], **Yang Lu**[1, 2, 3], **Yuan Xie**[4, 5*], **Yanyun Qu**[1, 2, 3*]

[1]School of Informatics, Xiamen University
[2]Institute of Artificial Intelligence, Xiamen University
[3]Key Laboratory of Multimedia Trusted Perception and Efficient Computing, Ministry of Education of China, Xiamen University
[4]School of Computer Science and Technology, East China Normal University
[5]Chongqing Institute of East China Normal University

Abstract

Domain Generalized Semantic Segmentation (DGSS) aims to utilize segmentation model training on known source domains to make predictions on unknown target domains. Currently, there are two kinds of network architectures: one based on Convolutional Neural Networks (CNNs) and the other based on Visual Transformers (ViTs). However, both CNN-based and ViT-based DGSS methods face challenges: the former lacks a global receptive field, while the latter requires more computational demands. Drawing inspiration from State Space Models (SSMs), which not only possess a global receptive field but also maintain linear complexity, we propose SSM-based method for achieving DGSS. In this work, we first elucidate why does *mask* make sense in SSM-based DGSS and propose our *mask* learning mechanism. Leveraging this mechanism, we present our *Mask Vision Mamba* network (MaskViM), a model for SSM-based DGSS, and design our *mask* loss to optimize MaskViM. Our method achieves superior performance on four diverse DGSS setting, which demonstrates the effectiveness of our method.

Introduction

Domain Generalized Semantic Segmentation (DGSS) (Ding et al. 2023; Ahn et al. 2024) aims to perform semantic segmentation on unknown target domains via models trained on known source domains. The essence of DGSS lies in enhancing the model's robustness against unknown target domains. To this end, there are two ways: (1) enabling the model to learn the target domain distribution (Peng et al. 2021; Yue et al. 2019; Lee et al. 2022; Wu et al. 2022), and (2) preventing the model from overfitting to the source domain distribution (Pan et al. 2018; Choi et al. 2021; Huang et al. 2021; Pan et al. 2019; Peng et al. 2022). The former way is often infeasible as the domain generalization paradigm restricts access to the target domain during training. Thus, adopting the latter way becomes a practical approach.

As many studies have shown (Hoyer, Dai, and Van Gool 2022; Paul and Chen 2022; Wenzel et al. 2022; Zhou et al. 2022; Zhang et al. 2024, 2022, 2021), improving the receptive field can greatly prevent the model from overfitting to the source domain. Thus, ViTs exhibit greater robustness than CNNs in out-of-distribution generalization due to their global receptive field. However, ViT-based methods incur higher computational demands. Recently, State Space Models (SSMs), represented by Mamba (Gu and Dao 2023), have achieved great success in visual fields (Gu, Goel, and Ré 2021; Liu et al. 2024). Compared with ViTs, Mamba not only possesses a global receptive field to improve model robustness but also maintains linear complexity to boost model efficiency. Therefore, SSM-based DGSS seems to be a more favorable choice.

However, we observe that directly applying SSMs to DGSS does not yield promising performance. There are two reasons: (1) Typically, SSM-based methods are inherently more suitable for 1-D causal data, such as text, rather than 2-D non-causal data like images (Liu et al. 2024; Zhu et al. 2024). Unlike text, where the 1-D order of words indicates causal relationships, the 2-D spatial positioning of pixels in

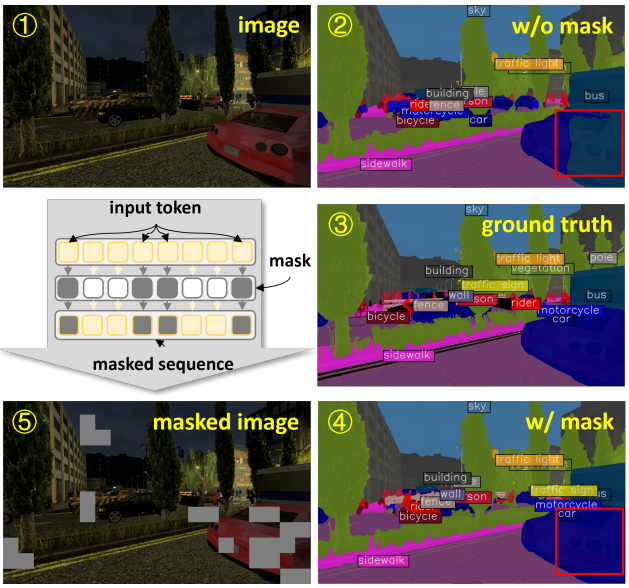

Figure 1: Visualization of the SSM-based DGSS both with and without *mask*. ① Original image. ② Prediction without *mask*. ③ Ground truth. ④ Prediction with *mask*. ⑤ Visualization of *mask* in original image.

*Corresponding author.

images describes such relationships. When unfolding images into 1-D sequences, the 2-D spatial relationships are disrupted, leading to tokens with incorrect positional relationships—referred to as non-causal tokens. This can obscure the semantic integrity of sequences, adversely affecting pixel-level prediction tasks. (2) In addition, we observe that SSM-based methods focus more on historical knowledge of the model, making previous source domain knowledge dominate the target domain reasoning, ultimately impairing the domain generalization performance.

In this work, we reveal the essential problems of SSM-based DGSS and conduct a toy experiment to demonstrate them, thereby proposing our *mask* learning mechanism. The mechanism locates and filters out the non-causal tokens via a learnable binary *mask*, which addresses the problems faced by SSM-based DGSS. In Fig. 1 ①, the bottom right corner displays a car positioned behind a bus. In Fig. 1 ②, indicated by the red box, SSM-based DGSS without *mask* disrupts the positional relationship between the car and the bus, resulting in misclassification of the car as the bus. In Fig. 1 ④, introducing the mechanism corrects the misclassification (indicated by the red box). In Fig. 1 ⑤, the occluded area represents non-causal tokens localized by this mechanism. Notably, the car is significantly obscured, indicating that SSM-based DGSS inaccurately captures the positional relationship in this region. Based on the mechanism, we introduce our *Mask Vision Mamba* network (MaskViM) for DGSS, which is a SSM-based encoder-decoder network, and design our *mask* loss to regulate the *mask* ratio and position effectively. Our contributions are summarized as follows:

- We propose MaskViM for DGSS to effectively mitigate the adverse effects of the non-causal tokens.
- We reveal the fundamental issues of SSM-based DGSS, and propose a novel *mask* learning mechanism. Furthermore, we devise a *mask* loss to control the *mask* ratio and position.
- We achieve superior performance on four DGSS settings, with an average increase of +1.9%, +1.7%, +1.2%, and + 1.7% mIoU compared to HGFormer, and +1.7% mIoU compared to BlindNet.

Related Works

Domain Generalization Semantic Segmentation DGSS primarily focuses on enhancing the model's generalization capability via randomization and normalization (Bi, You, and Gevers 2024). Randomization methods (Peng et al. 2021; Yue et al. 2019) aim to diversify the source domain distribution in hopes of encompassing the target domain distribution. For instance, Huang *et al.* (Huang et al. 2018) introduce an auxiliary domain to perform style transfer on the source domain at the data level to improve generalization. Lee *et al.* (Lee et al. 2022) utilize ImageNet data (Deng et al. 2009) as the auxiliary domain to conduct style transfer at the feature level. Wu *et al.* employ AdaIN (Huang and Belongie 2017) to extract color-jittered features and blend them with the original features to diversify the feature space. Normalization methods (Choi et al. 2021; Pan et al. 2018, 2019; Peng et al. 2022) aim to diminish the source domain style present in the features, thereby extracting domain-invariant content. Batch normalization (BN) (Ioffe and Szegedy 2015) preserves domain-invariant content information, and instance normalization (IN) (Ulyanov, Vedaldi, and Lempitsky 2016) reduces the impact of source domain style. Consequently, most methods explore how to integrate the benefits of both normalizations. For example, Pan *et al.* (Pan et al. 2019) were the first to propose switchable whitening, combining BN and IN. Whitening transformation, which standardizes features by decorrelating the channels, can also extract domain-invariant content. Luo (Luo 2017) and Cho *et al.* (Cho et al. 2019) apply whitening transformation in a group-wise manner. Choi *et al.* (Choi et al. 2021) enhance the approach to perform whitening transformation in an instance-selective mode.

State Space Models SSMs establish causal links for all states within sequences to predict unknown states, demonstrating significant potential in sequence modeling. Modeling for long sequences has consistently been a challenging issue. To address this, LSSL (Gu et al. 2021) integrates continuous state space models with HiPPO initialization (Gu et al. 2020) to mitigate the long sequence dependency challenge. The HiPPO matrix, capable of generating a hidden state that retains historical information, can manage long-range dependencies in both recurrent and convolutional representations. However, the method's intensive parameterization and substantial memory requirements render it impractical for widespread use. S4 (Gu, Goel, and Ré 2021) offers a solution that employs matrix factorization to reparameterize parameters into a diagonal structure, hence termed structured state space for sequences. Motivated by the concept of structured state space, numerous studies have concentrated on designing diverse structures for state space models, such as complex-diagonal structures (Gupta, Gu, and Berant 2022; Gu et al. 2022), support for multiple inputs and outputs (Smith, Warrington, and Linderman 2022), decomposition into diagonal plus low-rank operations (Hasani et al. 2022), and selection mechanisms (Gu and Dao 2023). These methods have achieved considerable success with long-range and causal data, such as language and speech (Mehta et al. 2022). However, there remains a scarcity of research on 2-D non-causal data, like images.

Mamba for Vision Task Mamba (Gu and Dao 2023), also known as the structured state space model with selection mechanism, has garnered significant attention since its introduction. Its performance slightly surpasses that of the Transformer with an equivalent parameter size, and it offers a computational advantage, being more than five times faster. Recent studies have validated Mamba's efficacy in visual tasks. In the field of visual representation learning, Vim (Zhu et al. 2024) proposes a novel generic vision backbone featuring bidirectional Mamba blocks, which mark image sequences with positional embeddings and compress visual representations using bidirectional state space models. VMamba (Liu et al. 2024) introduces a Swin-like hierarchical vision backbone and designs a 2D-selective-scan mechanism to transform 2-D non-causal images into 1-D sequences for Mamba-based modeling.

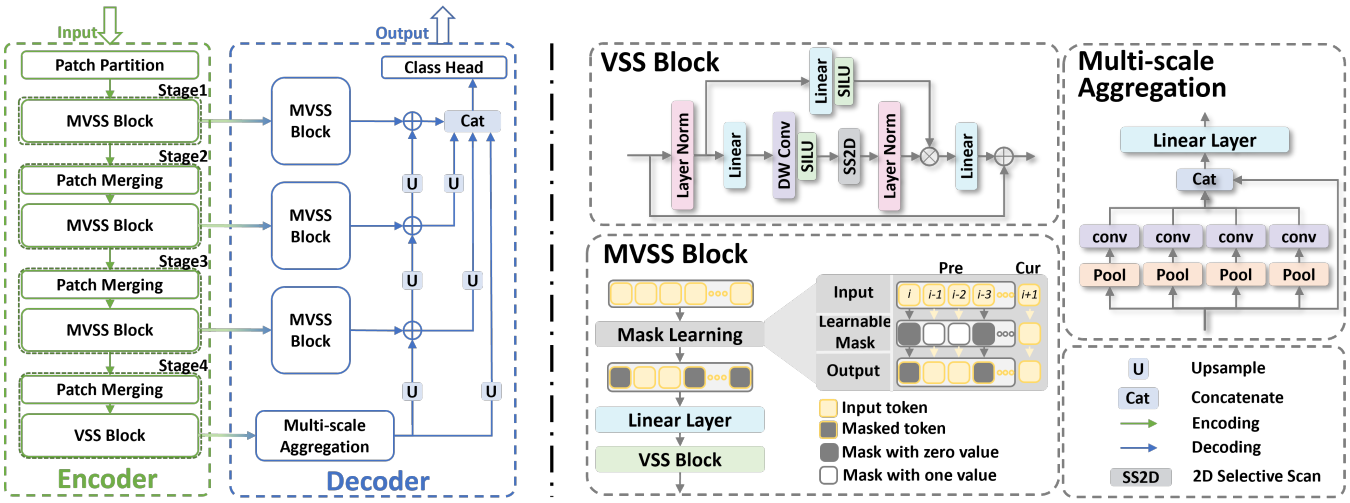

Figure 2: Overview of MaskViM. The network follows an encoder-decoder architecture: (1) the encoder network, which features four Stages with the initial three incorporating our MVSS block and the final Stage utilizing the VSS block; and (2) the decoder network, which integrates our MVSS block and a multi-scale aggregation module.

Preliminaries

Problem Definition Formally, we define the source and target domains as $\{\boldsymbol{I}^s, \boldsymbol{G}^s\}$ and $\{\boldsymbol{I}^t, \boldsymbol{G}^t\}$, respectively. Here, $\boldsymbol{I}$ denotes the input images and $\boldsymbol{G}$ denotes the corresponding ground truths. The superscript s and t indicate the source and target domains. Let $\boldsymbol{O}^s$ and $\boldsymbol{O}^t$ denote the segmentation predictions for the source and target domains, respectively. DGSS aims to obtain the target domain predictions $\boldsymbol{O}^t$ by training on $\boldsymbol{I}^s$ using their ground truth $\boldsymbol{G}^s$.

State Space Model SSM can be defined by the state equation $\boldsymbol{h}'(t)$ and the observation equation $y(t)$:

$$\begin{aligned} \boldsymbol{h}'(t) &= \boldsymbol{A}\boldsymbol{h}(t) + \boldsymbol{B}x(t), \quad x(t) \in \mathbb{R} \\ y(t) &= \boldsymbol{C}\boldsymbol{h}(t) + Dx(t), \quad y(t) \in \mathbb{R} \end{aligned} \tag{1}$$

where $\boldsymbol{A} \in \mathbb{R}^{d\times d}$, $\boldsymbol{B} \in \mathbb{R}^{d\times 1}$, $\boldsymbol{C} \in \mathbb{R}^{1\times d}$, $D \in \mathbb{R}$ and $\boldsymbol{h}(t) \in \mathbb{R}^{d\times 1}$.

Discrete SSM To apply SSM into deep learning algorithms, SSM transforms into its discrete version according to zero-order hold rule, i.e., $\overline{\boldsymbol{A}} = \exp(\Delta\boldsymbol{A}), \overline{\boldsymbol{B}} = \Delta\boldsymbol{B}, \Delta \in \mathbb{R}$, and can be defined as:

$$\begin{aligned} \boldsymbol{h}_{i+1} &= \overline{\boldsymbol{A}}\boldsymbol{h}_i + \overline{\boldsymbol{B}}x_i, \quad x_i \in \mathbb{R} \\ y_i &= \boldsymbol{C}\boldsymbol{h}_i + Dx_i, \quad y_i \in \mathbb{R} \end{aligned} \tag{2}$$

where $\overline{\boldsymbol{A}} \in \mathbb{R}^{d\times d}$, $\overline{\boldsymbol{B}} \in \mathbb{R}^{d\times 1}$, $\boldsymbol{C} \in \mathbb{R}^{1\times d}$, $D \in \mathbb{R}$ and $\boldsymbol{h}_i \in \mathbb{R}^{d\times 1}$.

Selective State Space Model Selective SSM, i.e., Mamba, utilizes linear and softplus to allow $\boldsymbol{B}, \boldsymbol{C}, \Delta$ to be $\boldsymbol{B}_i, \boldsymbol{C}_i, \Delta_i$, which are the functions of the input x_i and input-dependent. Thus, $\overline{\boldsymbol{A}}_i = \exp(\Delta_i\boldsymbol{A})$ and $\overline{\boldsymbol{B}}_i = \Delta_i\boldsymbol{B}_i$ are also input-dependent. The selective state space model can be defined as:

$$\begin{aligned} \boldsymbol{h}_{i+1} &= \overline{\boldsymbol{A}}_i\boldsymbol{h}_i + \overline{\boldsymbol{B}}_i x_i, \quad x_i \in \mathbb{R} \\ y_i &= \boldsymbol{C}_i\boldsymbol{h}_i + Dx_i, \quad y_i \in \mathbb{R} \end{aligned} \tag{3}$$

where $\overline{\boldsymbol{A}}_i \in \mathbb{R}^{d\times d}$, $\overline{\boldsymbol{B}}_i \in \mathbb{R}^{d\times 1}$, $\boldsymbol{C}_i \in \mathbb{R}^{1\times d}$, $D \in \mathbb{R}$ and $\boldsymbol{h}_i \in \mathbb{R}^{d\times 1}$. Given the parameter $\overline{\boldsymbol{A}}_i$ is diagonal and the parameters $\Delta_i, D \in \mathbb{R}$ (Han et al. 2024), Eq. 3 can be rewritten as:

$$\begin{aligned} \boldsymbol{h}_{i+1} &= \widetilde{\boldsymbol{A}}_i \odot \boldsymbol{h}_i + \boldsymbol{B}_i(\Delta_i \odot x_i), \quad x_i \in \mathbb{R} \\ y_i &= \boldsymbol{C}_i\boldsymbol{h}_i + D \odot x_i, \quad y_i \in \mathbb{R} \end{aligned} \tag{4}$$

where $\odot$ denotes the Hadamard product, $\widetilde{\boldsymbol{A}}_i = \mathrm{diag}(\overline{\boldsymbol{A}}_i) \in \mathbb{R}^{d\times 1}$, $\boldsymbol{B}_i \in \mathbb{R}^{d\times 1}$, $\boldsymbol{C}_i \in \mathbb{R}^{1\times d}$, $D \in \mathbb{R}$ and $\boldsymbol{h}_i \in \mathbb{R}^{d\times 1}$. Thus, the scalar output $y_{i+1} \in \mathbb{R}$ can be defined as:

$$y_{i+1} = \boldsymbol{C}_{i+1}[\widetilde{\boldsymbol{A}}_i \odot \boldsymbol{h}_i + \boldsymbol{B}_i(\Delta_i \odot x_i)] + D \odot x_{i+1} \tag{5}$$

To deal with an input sequence $\boldsymbol{x} \in \mathbb{R}^{N\times C}$, Eq. 5 can be rewritten as:

$$\boldsymbol{y}_{i+1} = \boldsymbol{C}_{i+1}[\widetilde{\boldsymbol{A}}_i \odot \boldsymbol{h}_i + \boldsymbol{B}_i(\boldsymbol{\Delta}_i \odot \boldsymbol{x}_i)] + \boldsymbol{D} \odot \boldsymbol{x}_{i+1} \tag{6}$$

where $\boldsymbol{y}_{i+1}, \boldsymbol{x}_i \in \mathbb{R}^{1\times C}$, $\boldsymbol{\Delta}_i \in \mathbb{R}^{1\times C}$, $\widetilde{\boldsymbol{A}}_i \in \mathbb{R}^{d\times C}$, $\boldsymbol{B}_i \in \mathbb{R}^{d\times 1}$, $\boldsymbol{C}_i \in \mathbb{R}^{1\times d}$, $D \in \mathbb{R}^{1\times C}$ and $\boldsymbol{h}_i \in \mathbb{R}^{d\times C}$.

Approach

Why Does *Mask* Make Sense in SSM-based DGSS? We start by discussing why utilizing *mask* in SSM is desirable for DGSS.

First, we explore the relationship between self-attention and SSM to reveal the problem of the SSM-based DGSS. Specifically, we denote $\boldsymbol{C}_i, \boldsymbol{B}_i, (\boldsymbol{\Delta}_i \odot \boldsymbol{x}_i)$ as $\boldsymbol{Q}_i, \boldsymbol{K}_i^\top, \boldsymbol{V}_i$. Given $\widetilde{\boldsymbol{A}}_i$ falls between 0 and 1, it can be viewed as the attention for the previous hidden state. Thus, we set $\overline{\boldsymbol{h}}_i = \widetilde{\boldsymbol{A}}_i \odot \boldsymbol{h}_i \in \mathbb{R}^{d\times C}$, and rewrite Eq. 6 as:

$$\boldsymbol{y}_{i+1} = \boldsymbol{Q}_{i+1}[\overline{\boldsymbol{h}}_i + \boldsymbol{K}_i^\top \boldsymbol{V}_i] + \boldsymbol{D} \odot \boldsymbol{x}_{i+1} \tag{7}$$

Given $\overline{\boldsymbol{h}}_i$ is equivalent to a function of the previous input tokens $\{\boldsymbol{x}_{i-1}, \boldsymbol{x}_{i-2}, ..., \boldsymbol{x}_0\}$, it effectively acts as a feature adapter of $\sum_{j=0}^{i-1} \boldsymbol{K}_j^\top \boldsymbol{V}_j$. Thus, we approximate $\overline{\boldsymbol{h}}_i \approx$

$\sum_{j=0}^{i-1} \boldsymbol{K}_j^\top \boldsymbol{V}_j$. In addition, due to $\boldsymbol{D}$ is input-independent, $\boldsymbol{D} \odot \boldsymbol{x}_{i+1}$ can be viewed as a shortcut of the current input $\boldsymbol{x}_{i+1}$. Therefore, we rewrite Eq. 7 as:

$$\boldsymbol{y}_{i+1} \approx \boldsymbol{Q}_{i+1}(\sum_{j=0}^{i} \boldsymbol{K}_j^\top \boldsymbol{V}_j) + \boldsymbol{D} \odot \boldsymbol{x}_{i+1} \tag{8}$$

This indicates that the current output $\boldsymbol{y}_{i+1}$ consists of the self-attention ($\boldsymbol{Q}_{i+1}(\sum_{j=0}^{i} \boldsymbol{K}_j^\top \boldsymbol{V}_j)$) of the previous inputs $\{\boldsymbol{x}_i, \boldsymbol{x}_{i-1}, ..., \boldsymbol{x}_0\}$, which is viewed as the predictions of the previous inputs, and the shortcut ($\boldsymbol{D} \odot \boldsymbol{x}_{i+1}$) of the current input $\boldsymbol{x}_{i+1}$, which is viewed as the offset of the current input. This means that the prediction of the current input is based on the previous inputs' predictions, with an added offset associated with the current input (since the current input does not participate in self-attention computation but follows a shortcut mode). Thus, if the previous inputs differ greatly from the current ones, i.e., there is a lack of causal relationship between them, the current output is highly susceptible to being influenced by previous inputs, resulting in predictions that are more aligned with the previous inputs and less relevant to the current input. This usually arises in DGSS due to that (1) significant domain discrepancies make the previous knowledge of the source domain dominate the reasoning of the current target domain, and (2) pixel-level predictions for the current rare tokens are highly susceptible to those for previous common tokens. Therefore, it is necessary to use *mask* to control the impact of previous non-causal tokens on current reasoning.

Second, we conduct a toy experiment to illustrate the importance of *mask* in SSM-based DGSS. We visualize the activation map of $\boldsymbol{Q}\boldsymbol{K}^\top$ from VMamba both with and without *mask*. As shown in Fig. 3 (b), the results from VMamba without *mask* show that the tokens indicated by red box tend to focus more on the common tokens in the previous inputs (even if they belong to different classes), while paying less attention to those of the same class in the current input. As shown in Fig. 3 (c), utilizing *mask* make the tokens indicated by red box more relevant to current token of same class and less relevant to previous tokens of different classes.

***Mask* Learning in SSM-based DGSS** We introduce how to utilize *mask* in SSM-based DGSS. Specifically, let $\boldsymbol{f}_{i,j}$ represent the token in SSMs (i.e., 2-D expression of $\boldsymbol{x}_i$), and $\boldsymbol{m}_{i,j} \in \{0, 1\}$ denote discrete binary *mask*, where i and j correspond to the horizontal and vertical coordinates, respectively. If $\boldsymbol{m}_{i,j} = 0$, the token $\boldsymbol{f}_{i,j}$ is excluded. Thus, the token $\boldsymbol{f}_{i,j}$ with *mask* is updated as:

$$\boldsymbol{f}_{i,j} = \begin{cases} \boldsymbol{f}_{i,j}, & \text{if } \boldsymbol{m}_{i,j} = 1; \\ 0, & \text{otherwise.} \end{cases} \tag{9}$$

To obtain an effective *mask* $\boldsymbol{m}_{i,j}$, a naive approach involves random generation following a uniform distribution, which is defined as:

$$\boldsymbol{m}_{i,j} = \textbf{onehot}(\max\{k | \sum_{k=0}^{1} p_{i,j}^k \leq u\}) \tag{10}$$

where u is sampled from a standard uniform distribution $\mathcal{U}(0, 1)$, and $p_{i,j}^k$ denotes the likelihood of assigning $\boldsymbol{m}_{i,j}$ to class $k \in \{0, 1\}$. Eq. 10 indicates that $\boldsymbol{m}_{i,j}$ selects class k when the cumulative probability up to k exceeds u. But this approach is sub-optimal due to the varying suitability of each positions for *mask*. Thus, a learnable *mask* $\boldsymbol{m}_{i,j}$ tailored to each token $\boldsymbol{f}_{i,j}$ is necessary.

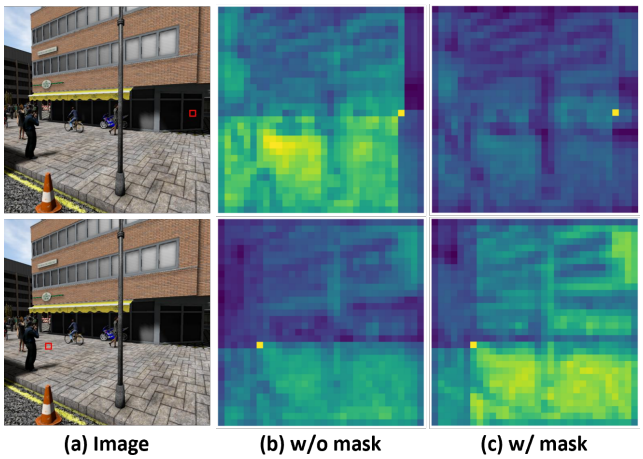

Figure 3: Visualization of the activation map of $\boldsymbol{Q}\boldsymbol{K}^\top$ for tokens indicated by red box. (a) Original image. (b) Results without *mask*. (c) Results with *mask*.

However, the binary nature of $\boldsymbol{m}_{i,j}$–limited to values of 0 or 1–poses a challenge for differentiability during backpropagation. This makes $\boldsymbol{m}_{i,j}$ not to be learnable. Given that *mask* learning process is equivalent to random sampling from a uniform distribution, we adopt the Gumbel-Max trick (Jang, Gu, and Poole 2016) to address the challenge. Thus, we rewrite Eq. 10 as:

$$\boldsymbol{m}_{i,j} = \textbf{onehot}(\arg\max_k \{g^k + \log p_{i,j}^k\}) \tag{11}$$

where $g^k = -\log(-\log(u))$ represents the Gumbel perturbation. However, the argmax operation is still non-differentiable. Thus, we employ the Gumbel-Softmax technique (Xu et al. 2022a; Xue et al. 2022) to replace the argmax operation with the Softmax function, and modify Eq. 11 as:

$$\boldsymbol{m}_{i,j} = \frac{\exp\left((\log(p_{i,j}^1) + g^1)/\tau\right)}{\sum_{k=0}^{1} \exp\left((\log(p_{i,j}^k) + g^k)/\tau\right)} \tag{12}$$

where $\tau > 0$ serves as the temperature parameter that modulates the Softmax function to approximate the argmax operation. Obviously, $p_{i,j}^k$ stands as the sole learnable parameter, allowing it to take any value within its range. Therefore, the objective of learning binary *mask* is achievable by tuning the differentiable variable $p_{i,j}^k$, which is initialized by sampling from a uniform distribution $\mathcal{U}(-\gamma, \gamma)$, with $\gamma > 0$ serving as a hyperparameter.

Mask Vision Mamba Network Based on our *mask* learning mechanism, we introduce our MaskViM network. As shown in Fig. 2, the input image undergoes hierarchical encoding to produce four multi-scale feature maps by the encoder. These maps are subsequently processed by the decoder to obtain the final segmentation prediction. Both encoder and decoder are composed of our MVSS block, which

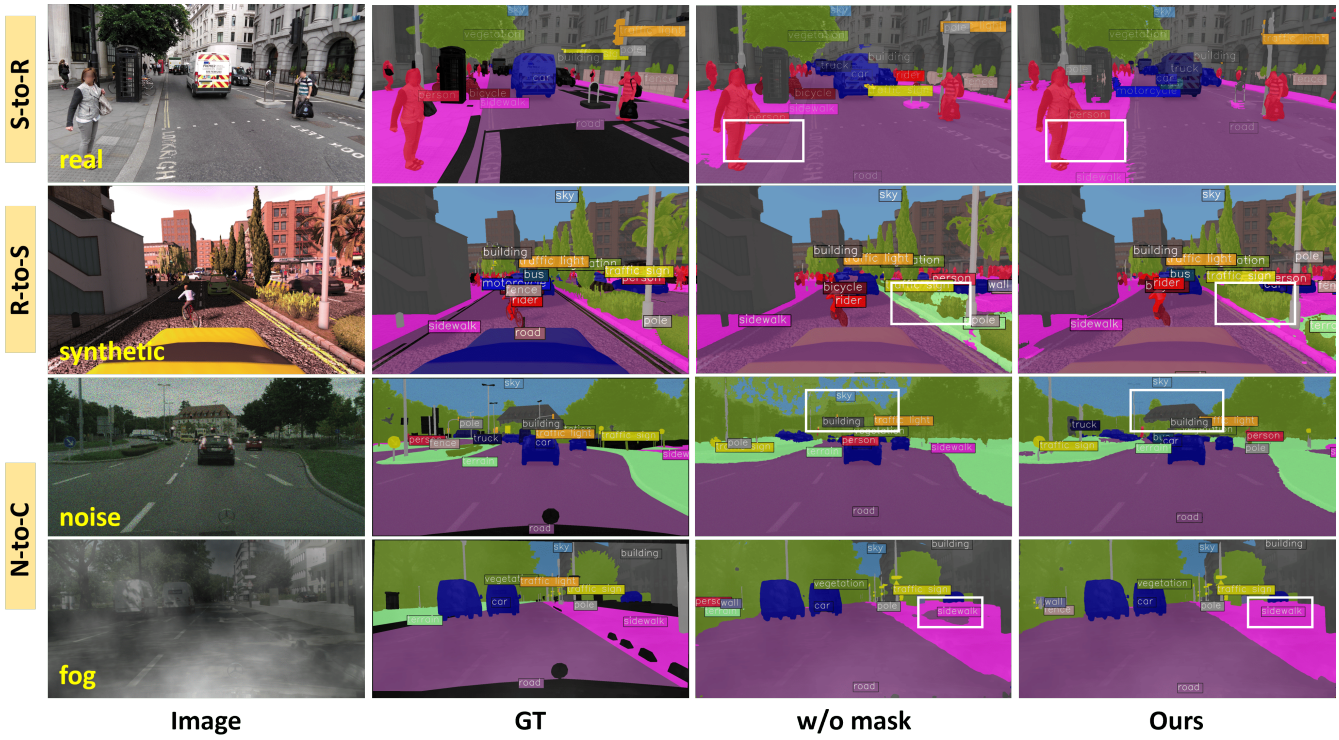

Figure 4: Visualizations of three kinds of DGSS scenarios: Synthetic-to-Real (S-to-R), Real-to-Synthetic (R-to-S), and Normal-to-Corruption (N-to-C) generalizations. The real and synthetic datasets are sourced from Mapillary and SYNTHIA validation sets, respectively. Corruptions, such as noise and fog, are introduced to Cityscapes validation set.

introduces the proposed *mask* learning mechanism into the VSS block (Liu et al. 2024) to locate and filter out previous non-causal tokens on current reasoning.

The encoder network generates hierarchical feature maps. The process begins by partitioning the input into 4×4 patches using the patch partition module, i.e., a 4×4 conv-norm layer with a stride of 4. These patches are then fed into four Stages to output their feature maps in turn. In Stages, the patch merging module (Liu et al. 2021) downsamples the feature maps. The outputs of each Stage serve as inputs to the decoder network. Note that the encoder architecture resembles the VMamba and the Swin Transformer (Liu et al. 2021), with the key difference being the replacement of their core block with our MVSS block.

The decoder network decodes semantic from the encoder to obtain predictions. The output of Stage 4 inputs into the multi-scale aggregation module, while those of the preceding three Stages fed into our MVSS block. The rationale for not utilizing the MVSS block in Stage 4 is straightforward: the output contains more semantic information but has lower resolution, meaning that masking even a few tokens could result in substantial semantic information loss. The multi-scale aggregation module is adopted akin to the pyramid pooling module in PSPNet (Zhao et al. 2017). Finally, the class head module maps the feature map into class space to obtain the segmentation prediction.

Optimization Objective Considering that masking too many tokens may lead to performance degradation and incorrect *mask* positioning can also impair performance, we devise a *mask* loss, i.e., $\mathcal{L}_m = \alpha_1 \cdot \mathcal{L}_m^r + \alpha_2 \cdot \mathcal{L}_m^p$, where α_1 and α_2 are weighting factors, to regulate both the ratio and position of *mask* as:

$$\begin{aligned} \mathcal{L}_m^r &= \frac{1}{S}\sum_{s=1}^{S}(\lambda - \frac{1}{h \cdot w}\sum_{i,j=0}^{h \cdot w} \boldsymbol{m}_{i,j}^s)^2, \\ \mathcal{L}_m^p &= \frac{1}{S}\sum_{s=1}^{S}\sum_{i,j=0}^{h \cdot w}(\boldsymbol{M}_{i,j} - \boldsymbol{m}_{i,j}^s)^2 \end{aligned} \tag{13}$$

where $\mathcal{L}_m^r$ and $\mathcal{L}_m^p$ represent the *mask* loss for the ratio and the position, respectively, and S represents the number of MVSS blocks. In this context, $\boldsymbol{m}_{i,j}^s$ denotes $\boldsymbol{m}_{i,j}$ in s-th MVSS block. The coefficient $\lambda \in (0, 1)$ modulates the *mask* ratio, and $\boldsymbol{M}_{i,j}$ controls the *mask* position, which is a binary label generated through element-wise comparison between the segmentation prediction and the ground truth. Note that the binary label map $\boldsymbol{M}$ is resized to match *mask* map $\boldsymbol{m}$

Methods	Backbone	#Params(M) ↓	FLOPs(G) ↓	mIoU
UperNet	Swin-T	60	945	47.1
UperNet	Swin-S	81	1039	50.3
UperNet	Swin-B	121	1188	50.9
MaskViM	Mask-T	60	305	56.4
MaskViM	Mask-S	82	346	59.1
MaskViM	Mask-B	118	390	59.9

Table 1: Efficiency comparison. FLOPs are computed for inputs of size 512×512, using the code from Mamba (Gu and Dao 2023). The mIoU is calculated on BDD validation set via models trained on Cityscapes training set.

Methods	Average	Blur				Noise				Digital				Weather			
		Motion	Defoc	Glass	Gauss	Gauss	Impul	Shot	Speck	Bright	Contr	Satur	JPEG	Snow	Spatt	Fog	Frost
Mask2former-T (Cheng et al. 2022)	41.6	51.5	49.4	38.2	46.2	9.6	9.8	13.5	44.4	74.2	60.0	70.0	23.3	23.7	59.4	65.4	27.3
HGFormer-T (Ding et al. 2023)	43.9	52.9	53.9	39.0	49.5	12.1	12.3	18.2	46.3	75.0	60.0	71.2	27.2	29.4	60.6	65.0	29.1
MaskViM-T	45.8	53.5	52.4	42.9	44.6	14.9	16.5	19.0	53.0	75.0	63.4	71.8	42.2	30.4	56.6	65.4	30.6

Table 2: Performance comparison (mIoU: %) in Cityscapes-to-Cityscapes-c setting. Red and blue fonts denote the best and second-best results, respectively.

Methods	Backbone	Trained on Cityscapes				
		B	M	G	S	Average
IBN (Pan et al. 2018)	ResNet50	48.6	57.0	45.1	26.1	44.2
SW (Pan et al. 2019)	ResNet50	48.5	55.8	44.9	26.1	43.8
DRPC (Yue et al. 2019)	ResNet50	49.9	56.3	45.6	26.6	44.6
RobustNet (Choi et al. 2021)	ResNet50	50.7	58.6	45.0	26.2	45.1
GTR (Peng et al. 2021)	ResNet50	50.8	57.2	45.8	26.5	45.0
WildNet (Lee et al. 2022)	ResNet50	50.9	58.8	47.0	28.0	46.2
ISW (Choi et al. 2021)	ResNet50	50.7	58.6	45.0	26.2	45.1
SiamDoGe (Wu et al. 2022)	ResNet50	51.5	59.0	45.1	26.7	45.6
DIRL (Xu et al. 2022b)	ResNet50	51.8	-	46.5	26.5	-
SAN-SAW (Peng et al. 2022)	ResNet50	53.0	59.8	47.3	28.3	47.1
BlindNet (Ahn et al. 2024)	ResNet50	51.8	60.2	48.0	28.5	47.1
Mask2former (Cheng et al. 2022)	ResNet50	46.8	61.6	48.0	31.2	46.9
HGFormer (Ding et al. 2023)	ResNet50	51.5	61.6	50.4	30.1	48.4
Mask2former (Cheng et al. 2022)	Swin-T	51.3	65.3	50.6	34.0	50.3
HGFormer (Ding et al. 2023)	Swin-T	53.4	66.9	51.3	33.6	51.3
MaskViM	Mask-T	56.4	66.6	54.3	34.6	53.0

Table 3: Performance comparison (mIoU: %) in Cityscapes-to-others setting.

via interpolation. The widely-adopted cross-entropy loss for semantic segmentation is denoted as $\mathcal{L}_{ce}$. Thus, the overall optimization objective is defined as:

$$\mathcal{L} = \mathcal{L}_{ce} + \mathcal{L}_{m} \tag{14}$$

Note that we adopt $\mathcal{L}_{m}^{r}$ to train the encoder and $\mathcal{L}_{m}^{p}$ to train the decoder, respectively.

Experiments

System Level Comparison

Efficiency We compare our approach with UperNet (Xiao et al. 2018) in efficiency as shown in Tab. 1. Despite having nearly identical #Params, our method achieves almost three times fewer FLOPs and a tenfold increase in mIoU.

Cityscapes-to-Cityscapes-c Tab. 2 presents the performance comparison in Cityscapes-to-Cityscapes-c setting. This setting enables us to evaluate our method's robustness. Our method surpasses HGFormer with an average increase of +1.9% mIoU. With the corruption of "Digital-JPEG", our approach achieves an impressive improvement of over +15% mIoU.

Cityscapes-to-Others Tab. 3 presents the performance comparison in the Cityscapes-to-others setting. Our method surpasses HGFormer by an average of +1.7% mIoU. For CNN-based methods, our approach shows a significant improvement, exceeding them by almost +5% mIoU. The real-to-synthetic generalization capability of our method is further evidenced in the C-to-G experiment, where it outperforms HGFormer by +3% mIoU.

Mapillary-to-Others Tab. 4 illustrates the performance comparison in the Mapillary-to-others setting. Our method outperforms HGFormer with an average increase of +1.2% mIoU, and nearly +3% mIoU over CNN-based methods.

Methods	Backbone	Trained on Mapillary				
		G	S	C	B	Average
IBN (Pan et al. 2018)	ResNet50	30.7	27.0	42.8	31.0	32.9
SW (Pan et al. 2019)	ResNet50	28.5	27.4	40.7	30.5	31.8
DRPC (Yue et al. 2019)	ResNet50	33.0	29.6	46.2	32.9	35.4
GTR (Peng et al. 2021)	ResNet50	32.9	30.3	45.8	32.6	35.4
ISW (Choi et al. 2021)	ResNet50	33.4	30.2	46.4	32.6	35.6
SAN-SAW (Peng et al. 2022)	ResNet50	34.0	31.6	48.7	34.6	37.2
Mask2former (Cheng et al. 2022)	ResNet50	55.8	37.7	65.6	56.4	53.9
HGFormer (Ding et al. 2023)	ResNet50	59.2	37.4	67.1	59.1	55.7
Mask2former (Cheng et al. 2022)	Swin-T	57.8	40.1	68.2	59.1	56.3
HGFormer (Ding et al. 2023)	Swin-T	60.1	39.5	69.3	61.0	57.5
MaskViM	Mask-T	60.8	38.8	73.1	62.0	58.7

Table 4: Performance comparison (mIoU: %) in Mapillary-to-others setting.

Methods	Backbone	Trained on GTAV				
		C	B	M	S	Average
IBN (Pan et al. 2018)	ResNet50	33.9	32.3	37.8	27.9	33.0
RobustNet (Choi et al. 2021)	ResNet50	37.3	35.2	40.3	28.3	35.3
WildNet (Lee et al. 2022)	ResNet50	44.6	38.4	46.1	31.3	40.1
SiamDoGe (Wu et al. 2022)	ResNet50	43.0	37.5	40.6	28.3	37.4
DIRL (Xu et al. 2022b)	ResNet50	41.0	39.2	41.6	-	-
SAN-SAW (Peng et al. 2022)	ResNet50	39.8	37.3	41.9	30.8	37.5
BlindNet (Ahn et al. 2024)	ResNet50	45.7	41.3	47.1	31.4	41.4
SHADE (Zhao et al. 2024)	ResNet50	44.7	39.3	43.3	-	-
HRDA(Hoyer, Dai, and Van Gool 2023)	MiT-B2	46.9	42.8	47.2	-	-
DAFormer (Hoyer, Dai, and Van Gool 2023)	MiT-B2	46.3	41.5	46.8	-	-
MaskViM	Mask-T	43.3	42.9	50.9	35.3	43.1

Table 5: Performance comparison (mIoU: %) in GTAV-to-others setting.

Mirroring the Cityscapes-to-others setting, our method continues to demonstrate exceptional generalization capability in real-to-synthetic scenarios.

GTAV-to-Others Tab. 5 depicts the performance comparison in the GTAV-to-others setting. This setting assess our method's synthetic-to-real scenarios adaptation. In comparison to BlindNet, our method excels with an average improvement of +1.7% mIoU. Specifically, in the G-to-M experiment, our method surpasses BlindNet by +3.8% mIoU.

Ablation Studies

***Mask* in the Encoder** We evaluate the impact of *mask* in the encoder, as shown in the first three lines of Tab. 6. We select VMamba+linear as baseline, and introduce *mask* into the blocks of the first three Stages ("+Block part.") and all Stages ("+Block all."), respectively. The setting of "+Block all." results in decreased performance (from 46.7% to 45.1%), suggesting that the feature map from the final Stage is not conducive to *mask* learning. We attribute this to the semantically rich yet lower-resolution nature of that map, where masking even a few tokens could lead to a significant loss of semantic information, thus diminishing overall performance. Thus, we select the setting of "+Block part." as our encoder design. In additional, we explore the impact of different *mask* ratios λ, as shown in Tab. 7. The

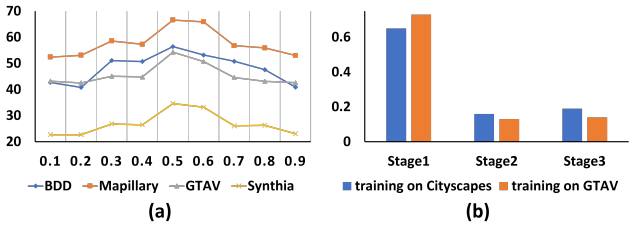

Figure 5: Analysis on *mask* in the encoder. (a) Performance comparison with different uniform λ. (b) *mask* ratios at the first three Stages with a uniform $\lambda = 0.5$.

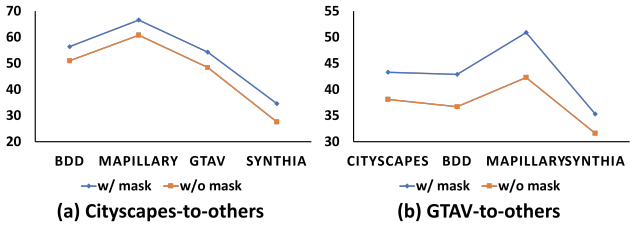

Figure 6: Performance comparison both with and without *mask*. (a) The Cityscapes-to-others setting. (b) The GTAV-to-others setting.

component "*mask* ratio λ at each Stage" refers to individual adjustment of λ to modulate the ratio at each Stage. The results indicate that a uniform λ for all Stages is better than an individual one for each Stage. Finally, we employ a uniform $\lambda = 0.5$ to regulate the ratio across all Stages. Fig. 5 (a) illustrates the effects of different uniform λ, with the best performance occurring at $\lambda = 0.5$. Fig. 5 (b) displays the ratio across the first three Stages of the encoder using the optimal $\lambda = 0.5$. The results highlight that non-causal tokens are prevalent in shallow layers; however, our *mask* learning mechanism effectively addresses these tokens, thereby mitigating their occurrence in deeper layers.

***Mask* in the Decoder** We evaluate the impact of *mask* in the decoder in the last three lines of Tab. 6. We denote Mask$_D$ as the decoder of our MaskViM. Let the setting of "*-mask*" denote the removal of the *mask* in Mask$_D$. The removal make performance decrease 2.5% mIoU (from 53.0% to 50.5%). Let the setting of "-All." denote the replacement of the multi-scale aggregation module with the VSS block based on the setting of "*-mask*". This results in a 1.3% mIoU drop (from 50.5% to 49.2%). These results highlight the beneficial role of *mask* in the decoder. In addition, we explore different depths of Mask$_D$ in Tab. 7. The component "Depths of Mask$_D$" refers to the number of MVSS blocks in the first three Stages of Mask$_D$. The results show that setting the component to (1,1,1) is optimal. Fig. 6 presents the performance comparison both with and without *mask* under Cityscapes-to-others and GTAV-to-others settings, further demonstrating the efficacy of *mask* in SSM-based DGSS.

Loss Analysis To demonstrate the importance of the *mask* loss $\mathcal{L}_m^r$ and $\mathcal{L}_m^p$, we contrast three different loss designs in Tab. 8. The combination $\mathcal{L}_{ce}$+$\mathcal{L}_m^r$ indicates regulation of the *mask* ratio. Conversely, $\mathcal{L}_{ce}$+$\mathcal{L}_m^p$ refers to the regulation of the *mask* position. The results highlight the necessity of managing both the *mask* ratio and position to optimize performance. Additionally, we examine the effects of the weighting factors α_1 and α_2, as shown in Tab. 7. The optimal weighting strategy occurs when $\alpha_1, \alpha_2 = 1.0$.

Encoder	Decoder	B	M	G	S	Average
VMamba	Linear	50.9	61.3	47.2	27.4	46.7
+Block all.	Linear	49.6	59.5	45.8	25.2	45.1
+Block part.	Linear	**51.7**	**62.9**	**48.6**	**29.1**	**48.1**
+Block part.	Mask$_D$	**56.4**	**66.6**	**54.3**	**34.6**	**53.0**
+Block part.	*-mask*	53.7	65.1	50.7	32.4	50.5
+Block part.	-All.	52.6	63.7	49.8	30.5	49.2

Table 6: Performance impact (mIoU: %) of *mask* in the encoder and decoder.

Component	Variant	B	M	G	S	Average
***mask* ratio λ at each** Stage	(0.9, 0.7, 0.5)	54.7	64.1	48.9	31.6	49.8
	(0.8, 0.6, 0.4)	52.3	65.4	46.5	27.6	48.0
	(0.5, 0.7, 0.9)	48.2	57.3	44.7	23.0	43.3
	(0.4, 0.6, 0.8)	47.9	56.8	44.5	24.1	43.3
	(0.5, 0.5, 0.5)	48.3	58.1	45.4	25.6	44.4
Depths of Mask$_D$	(2,2,2)	56.1	65.9	51.8	34.2	52.0
	(1,2,2)	**56.5**	**66.9**	51.7	34.1	52.3
	(1,1,2)	56.3	66.3	51.8	34.3	52.2
α_1 **and** α_2	(20.0, 10.0)	53.8	63.3	49.7	31.5	49.6
	(10.0,5.0)	55.1	65.7	51.3	33.7	51.4
	(5.0, 1.0)	55.5	**66.9**	51.7	34.1	52.1
MaskViM	0.5‖(1,1,1)‖(1.0, 1.0)	56.4	66.6	**54.3**	**34.6**	**53.0**

Table 7: Performance comparison (mIoU: %) of different components.

Loss	B	M	G	S	Average
$\mathcal{L}_{ce}$	40.8	50.7	36.7	19.1	36.8
$\mathcal{L}_{ce}$+$\mathcal{L}_m^r$	49.1	58.6	45.3	29.2	45.6
$\mathcal{L}_{ce}$+$\mathcal{L}_m^p$	48.6	59.1	44.5	28.4	45.2
$\mathcal{L}_{ce}$+$\mathcal{L}_m^r$+$\mathcal{L}_m^p$	**56.4**	**66.6**	**54.3**	**34.6**	**53.0**

Table 8: Performance comparison (mIoU: %) of different loss combinations.

Conclusion

In this work, we find that directly applying SSMs to DGSS does not yield promising performance and reveal that SSM-based DGSS will create the non-causal tokens that dominate the reasoning of the current input. To address this issue, we propose *mask* learning mechanism and introduce it into our MaskViM network for DGSS. We evaluate our method across four diverse DGSS settings, demonstrating that our method achieves superior performance.

Acknowledgements

This work is supported by the NSFC (62176224, U2268217, 62176092, 62222602, 62106075, 62476090, 62376233, 62431004); Natural Science Foundation of Shanghai (23ZR1420400); Natural Science Foundation of Chongqing (CSTB2023NSCQ-JQX0007); Natural Science Foundation of Fujian Province under Grant 2024J09001; China Computer Federation Lenovo Blue Ocean Research Fund; China Academy of Railway Sciences (2023YJ357); Xiaomi Young Talents Program.

References

Ahn, W.-J.; Yang, G.-Y.; Choi, H.-D.; and Lim, M.-T. 2024. Style Blind Domain Generalized Semantic Segmentation via Covariance Alignment and Semantic Consistence Contrastive Learning. *arXiv preprint arXiv:2403.06122*.

Bi, Q.; You, S.; and Gevers, T. 2024. Learning content-enhanced mask transformer for domain generalized urban-scene segmentation. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 38, 819–827.

Cheng, B.; Misra, I.; Schwing, A. G.; Kirillov, A.; and Girdhar, R. 2022. Masked-attention mask transformer for universal image segmentation. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 1290–1299.

Cho, W.; Choi, S.; Park, D. K.; Shin, I.; and Choo, J. 2019. Image-to-image translation via group-wise deep whitening-and-coloring transformation. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 10639–10647.

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 *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 11580–11590.

Deng, J.; Dong, W.; Socher, R.; Li, L.-J.; Li, K.; and Fei-Fei, L. 2009. Imagenet: A large-scale hierarchical image database. In *2009 IEEE conference on computer vision and pattern recognition*, 248–255. Ieee.

Ding, J.; Xue, N.; Xia, G.-S.; Schiele, B.; and Dai, D. 2023. Hgformer: Hierarchical grouping transformer for domain generalized semantic segmentation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 15413–15423.

Gu, A.; and Dao, T. 2023. Mamba: Linear-time sequence modeling with selective state spaces. *arXiv preprint arXiv:2312.00752*.

Gu, A.; Dao, T.; Ermon, S.; Rudra, A.; and Ré, C. 2020. Hippo: Recurrent memory with optimal polynomial projections. *Advances in neural information processing systems*, 33: 1474–1487.

Gu, A.; Goel, K.; Gupta, A.; and Ré, C. 2022. On the parameterization and initialization of diagonal state space models. *Advances in Neural Information Processing Systems*, 35: 35971–35983.

Gu, A.; Goel, K.; and Ré, C. 2021. Efficiently modeling long sequences with structured state spaces. *arXiv preprint arXiv:2111.00396*.

Gu, A.; Johnson, I.; Goel, K.; Saab, K.; Dao, T.; Rudra, A.; and Ré, C. 2021. Combining recurrent, convolutional, and continuous-time models with linear state space layers. *Advances in neural information processing systems*, 34: 572–585.

Gupta, A.; Gu, A.; and Berant, J. 2022. Diagonal state spaces are as effective as structured state spaces. *Advances in Neural Information Processing Systems*, 35: 22982–22994.

Han, D.; Wang, Z.; Xia, Z.; Han, Y.; Pu, Y.; Ge, C.; Song, J.; Song, S.; Zheng, B.; and Huang, G. 2024. Demystify Mamba in Vision: A Linear Attention Perspective. *arXiv preprint arXiv:2405.16605*.

Hasani, R.; Lechner, M.; Wang, T.-H.; Chahine, M.; Amini, A.; and Rus, D. 2022. Liquid structural state-space models. *arXiv preprint arXiv:2209.12951*.

Hoyer, L.; Dai, D.; and Van Gool, L. 2022. Daformer: Improving network architectures and training strategies for domain-adaptive semantic segmentation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 9924–9935.

Hoyer, L.; Dai, D.; and Van Gool, L. 2023. Domain adaptive and generalizable network architectures and training strategies for semantic image segmentation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*.

Huang, J.; Guan, D.; Xiao, A.; and Lu, S. 2021. Fsdr: Frequency space domain randomization for domain generalization. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 6891–6902.

Huang, X.; and Belongie, S. 2017. Arbitrary style transfer in real-time with adaptive instance normalization. In *Proceedings of the IEEE international conference on computer vision*, 1501–1510.

Huang, X.; Liu, M.-Y.; Belongie, S.; and Kautz, J. 2018. Multimodal unsupervised image-to-image translation. In *Proceedings of the European conference on computer vision (ECCV)*, 172–189.

Ioffe, S.; and Szegedy, C. 2015. Batch normalization: Accelerating deep network training by reducing internal covariate shift. In *International conference on machine learning*, 448–456. pmlr.

Jang, E.; Gu, S.; and Poole, B. 2016. Categorical reparameterization with gumbel-softmax. *arXiv preprint arXiv:1611.01144*.

Lee, S.; Seong, H.; Lee, S.; and Kim, E. 2022. WildNet: Learning domain generalized semantic segmentation from the wild. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 9936–9946.

Liu, Y.; Tian, Y.; Zhao, Y.; Yu, H.; Xie, L.; Wang, Y.; Ye, Q.; and Liu, Y. 2024. Vmamba: Visual state space model. *arXiv preprint arXiv:2401.10166*.

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.

Luo, P. 2017. Learning deep architectures via generalized whitened neural networks. In *International Conference on Machine Learning*, 2238–2246. PMLR.

Mehta, H.; Gupta, A.; Cutkosky, A.; and Neyshabur, B. 2022. Long range language modeling via gated state spaces. *arXiv preprint arXiv:2206.13947*.

Pan, X.; Luo, P.; Shi, J.; and Tang, X. 2018. Two at once: Enhancing learning and generalization capacities via ibn-net. In *Proceedings of the european conference on computer vision (ECCV)*, 464–479.

Pan, X.; Zhan, X.; Shi, J.; Tang, X.; and Luo, P. 2019. Switchable whitening for deep representation learning. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 1863–1871.

Paul, S.; and Chen, P.-Y. 2022. Vision transformers are robust learners. In *Proceedings of the AAAI conference on Artificial Intelligence*, volume 36, 2071–2081.

Peng, D.; Lei, Y.; Hayat, M.; Guo, Y.; and Li, W. 2022. Semantic-aware domain generalized segmentation. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2594–2605.

Peng, D.; Lei, Y.; Liu, L.; Zhang, P.; and Liu, J. 2021. Global and local texture randomization for synthetic-to-real semantic segmentation. *IEEE Transactions on Image Processing*, 30: 6594–6608.

Smith, J. T.; Warrington, A.; and Linderman, S. W. 2022. Simplified state space layers for sequence modeling. *arXiv preprint arXiv:2208.04933*.

Ulyanov, D.; Vedaldi, A.; and Lempitsky, V. 2016. Instance normalization: The missing ingredient for fast stylization. *arXiv preprint arXiv:1607.08022*.

Wenzel, F.; Dittadi, A.; Gehler, P.; Simon-Gabriel, C.-J.; Horn, M.; Zietlow, D.; Kernert, D.; Russell, C.; Brox, T.; Schiele, B.; et al. 2022. Assaying out-of-distribution generalization in transfer learning. *Advances in Neural Information Processing Systems*, 35: 7181–7198.

Wu, Z.; Wu, X.; Zhang, X.; Ju, L.; and Wang, S. 2022. SiamDoGe: Domain generalizable semantic segmentation using siamese network. In *European Conference on Computer Vision*, 603–620. Springer.

Xiao, T.; Liu, Y.; Zhou, B.; Jiang, Y.; and Sun, J. 2018. Unified perceptual parsing for scene understanding. In *Proceedings of the European conference on computer vision (ECCV)*, 418–434.

Xu, J.; De Mello, S.; Liu, S.; Byeon, W.; Breuel, T.; Kautz, J.; and Wang, X. 2022a. Groupvit: Semantic segmentation emerges from text supervision. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 18134–18144.

Xu, Q.; Yao, L.; Jiang, Z.; Jiang, G.; Chu, W.; Han, W.; Zhang, W.; Wang, C.; and Tai, Y. 2022b. Dirl: Domain-invariant representation learning for generalizable semantic segmentation. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 36, 2884–2892.

Xue, M.; Zhang, H.; Song, J.; and Song, M. 2022. Meta-attention for vit-backed continual learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 150–159.

Yue, X.; Zhang, Y.; Zhao, S.; Sangiovanni-Vincentelli, A.; Keutzer, K.; and Gong, B. 2019. Domain randomization and pyramid consistency: Simulation-to-real generalization without accessing target domain data. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2100–2110.

Zhang, Y.; Hu, R.; Li, R.; Qu, Y.; Xie, Y.; and Li, X. 2024. Cross-Modal Match for Language Conditioned 3D Object Grounding. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 38, 7359–7367.

Zhang, Y.; Qu, Y.; Xie, Y.; Li, Z.; Zheng, S.; and Li, C. 2021. Perturbed self-distillation: Weakly supervised large-scale point cloud semantic segmentation. In *Proceedings of the IEEE/CVF international conference on computer vision*, 15520–15528.

Zhang, Y.; Xie, Y.; Li, C.; Wu, Z.; and Qu, Y. 2022. Learning all-in collaborative multiview binary representation for clustering. *IEEE Transactions on Neural Networks and Learning Systems*, 35(3): 4260–4273.

Zhao, H.; Shi, J.; Qi, X.; Wang, X.; and Jia, J. 2017. Pyramid scene parsing network. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2881–2890.

Zhao, Y.; Zhong, Z.; Zhao, N.; Sebe, N.; and Lee, G. H. 2024. Style-hallucinated dual consistency learning: A unified framework for visual domain generalization. *International Journal of Computer Vision*, 132(3): 837–853.

Zhou, D.; Yu, Z.; Xie, E.; Xiao, C.; Anandkumar, A.; Feng, J.; and Alvarez, J. M. 2022. Understanding the robustness in vision transformers. In *International Conference on Machine Learning*, 27378–27394. PMLR.

Zhu, L.; Liao, B.; Zhang, Q.; Wang, X.; Liu, W.; and Wang, X. 2024. Vision mamba: Efficient visual representation learning with bidirectional state space model. *arXiv preprint arXiv:2401.09417*.