SPG-GT: Structural Prior Guided GNN-Transformers for Ship Landmark Detection

Mingxin Zhang, Guangbo Sun, Qian Zhang, Lin Zhang, Youmei Zhang, Paul L. Rosin, Ran Song*, and Wei Zhang

***Abstract*—The visual perception of ships has gained increasing attention in computer vision and ocean engineering. Ship landmark detection plays a crucial role in various applications, including ship recognition, ship image generation, key area detection, and ship detection. However, existing methods did not fully leverage the association among landmarks, which leads to a lack of overall perception of ships and limits the performance of ship landmark detection. To address this issue, this paper proposes SPG-GT, a ship landmark detection model that combines graph neural networks (GNNs) and Transformers guided by ship structural prior. GNNs effectively encode the connectivity information between ship landmarks, a set of keypoints important for defining the overall structure of a ship. SPG-GT also leverages Transformers and coordinate convolution to extract global and local features of a ship, which ensures that the detected landmarks are consistent with the nature of a ship. We evaluate SPG-GT on the publicly available SLAD dataset and a newly created SLAD++ dataset. The experimental results on both datasets demonstrate the superior performance of SPG-GT. Project web page: https://vsislab.github.io/SPGGT/.**

***Index Terms*—Ship Images, Landmark Detection, Maritime Transportation, Computer Vision**

I. Introduction

In recent years, the installation of cameras in ship berths or on board ships has become a common practice. Since monitoring multiple video streams for a long time is beyond human capability, an increasing number of researchers have focused on various tasks of ship surveillance through computer vision and machine learning techniques which has the advantage of low cost, high efficiency, and stable performance. Such tasks include ship recognition [1], [2], ship object detection [3], ship tracking [4], ship re-identification [5], and ship collision warning [6]. SLAD [7], the first ship dataset with landmark annotations, showed that ship landmarks have a range of applications, including ship recognition, ship image generation, key area detection for ships, and ship detection.

This paper focuses on the detection of ship landmarks. Recently, state-of-the-art methods such as CPM [8], HRNet [9] and ViTPose [10] showed a good performance in common keypoint or landmark detection tasks, such as those related to human bodies and vehicles. However, these methods are not specifically tailored for ship landmark detection [11]. Considering that a ship is a rigid object with a regular topological structure, there is a strong geometric correlation among its landmarks. Leveraging this correlation can enhance the overall perception of the ship and improve landmark detection performance. However, due to factors such as changes in viewpoint and ship movement, the visual associations among the landmarks on the ship exhibit a high degree of uncertainty, making it challenging to extract landmark association information from ship images. To address this issue, we introduce SPG-GT, a combination of graph neural networks (GNNs) and Transformers guided by ship structural prior for landmark detection.

Specifically, a GNN in SPG-GT takes the learned region of interest for each landmark as a node and establishes a connection matrix based on the ship structural prior. Then, SPG-GT aggregates the information from each node and its neighboring nodes, and updates node representations using the connectivity in the graph structure. This process enhances the representational capacity of the nodes, enabling the capture of the important information related to the ship structure. Next, SPG-GT leverages Transformer [12] to extract global features of ship landmarks. Note that Transformer possesses a strong capability of capturing global information in both natural language processing [13], [14] and computer vision [15], [16]. However, Transformer lacks the capability of local feature extraction and it is not efficient to directly incorporate Transformer into ship landmark detection. Thus, SPG-GT employs coordinate convolutions to preserve local information of landmarks in the process of feature learning. In SPG-GT, the GNNs which encode the correlation between local nodes, subject to the ship structure, further integrate local and global information for efficient landmark detection.

In summary, this work has the following main contributions:

1) we propose SPG-GT, a GNN-Transformer architecture for ship landmark detection. It leverages the prior knowledge of ship structure to guide the construction of a graph used for encoding the association among landmarks.
2) We establish a new ship image dataset, namely SLAD++. Compared to the SLAD dataset, it adds extra variations in weather (e.g. snowy, rainy, etc.) and time (e.g. sunrise, sunset, etc.) into ship images to ensure a good diversity of the data.
3) Through qualitative and quantitative experiments on

Mingxin Zhang, Guangbo Sun, Qian Zhang, Lin Zhang, Ran Song, and Wei Zhang are with the School of Control Science and Engineering at Shandong University, China (Email: 95zhangmingxin@gmail.com; 202214832@mail.sdu.edu.cn; zhq9669@gmail.com; zl935546110@gmail.com; zhangyoumei@qlu.edu.cn; ransong@sdu.edu.cn).

Youmei Zhang is with the School of Mathematics and Statistics, Qilu University of Technology(Shandong Academy of Sciences), China (Email: zhangyoumei@qlu.edu.cn).

Paul L. Rosin is with the School of Computer Science and Informatics at Cardiff University, UK (Email: RosinPL@cardiff.ac.uk).

*Corresponding author: Ran Song (Email: ransong@sdu.edu.cn).

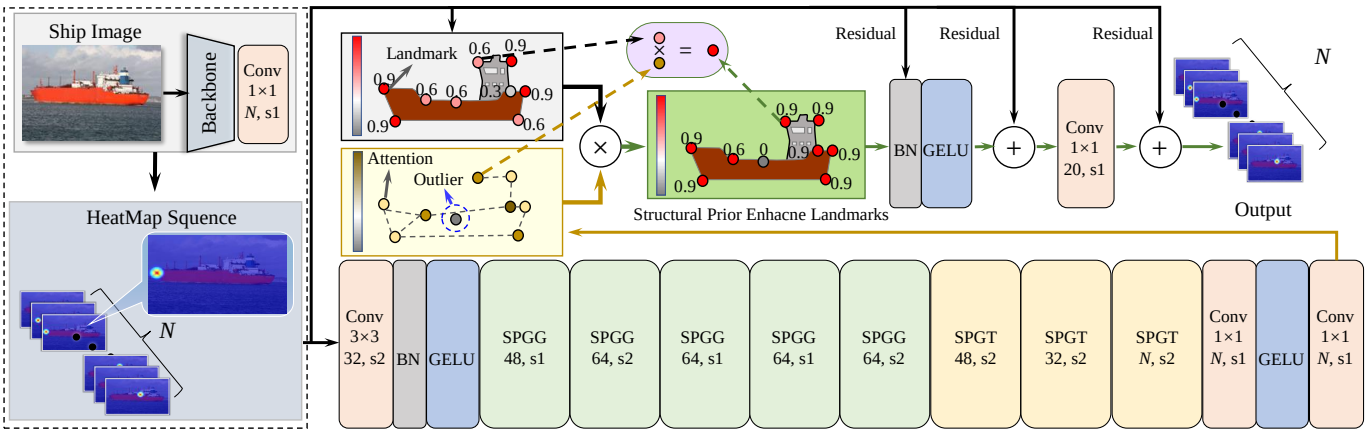


Fig. 1. Overall framework of SPG-GT. First, ship images are used to predict landmark heatmaps through a backbone feature extractor and a convolutional layer. Then, by using landmark heatmaps as input, SPG-GT leverages ship structural priors to enhance attention and explore inter-landmark dependencies. Finally, through SPGG and SPGT blocks, SPG-GT refines landmark prediction by enhancing the scores of inliers and suppressing those of outliers. BN represents the batch normalization layer, and GELU denotes the activation layer. 's1' denotes that the stride is 1.

both SLAD and SLAD++, we demonstrate that the proposed SPG-GT achieves the state-of-the-art performance on ship landmark detection.

II. Related Work

Landmark detection aims to identify specific local regions of interest in an image and has a wide range of applications [17]–[23]. It can broadly be classified into bottom-up and top-down methods.

A. Bottom-up Methods for Landmark Detection

Bottom-up methods begin with the detection of all landmarks across the entire image, and then employ a landmark clustering process to derive the final detection results. Insafutdinov et al. [24] proposed DeeperCut, featuring a more robust part detector and enhanced incremental optimization. Sheikh et al. [25] introduced part affinity fields with a nonparametric representation, aiming to associate body parts with individuals in images. Schiele et al. [26] expedited computation by simplifying the body-part relationship graph and leveraging a feed-forward convolutional architecture. Gall et al. [27] proposed a novel approach that combined multi-person pose estimation and tracking by representing body joint detections in videos with a spatio-temporal graph. Deng et al. [28] introduced associative embedding, presenting a method that instructed a network to output detections and group assignments concurrently. Ouyang et al. [29] introduced multi-person joint detector with multi-scale features for multi-scale feature extraction, effectively reducing false alarms by integrating global context. Murphy et al. [30] introduced a box-free bottom-up approach where keypoints were detected and their relative displacements predicted to group them into person pose instances. Cao et al. [31] pioneered a method using dual-channel predictions for Part Confidence Maps and Part Affinity Fields, enabling Multi-Person Parsing by transforming the problem into a graph-based approach and utilizing the Hungarian Algorithm for matching. Qian et al. [28], [32] introduced SpatialNet for body part association and TemporalNet for temporal consistency, predicting keypoint embedding, spatial instance embedding, human embedding, and temporal instance embedding. In prior bottom-up research, predicting region locations and assigning them to the same entity requires high-quality training data with multiple objects in a single image.

B. Top-down Methods for Landmark Detection

Top-down methods typically utilize the object's bounding box as prior knowledge, enabling the detector to concentrate on the object instead of the cluttered background. Yaser et al. [8] proposed Convolutional Pose Machines, utilizing Heatmaps to depict keypoint positions and their spatial constraints, while concurrently transmitting Heatmaps and Feature Maps through the network. Lu et al. [33] introduced a pioneering RMPE framework that effectively tackled issues arising from imprecise human bounding boxes. Girshick et al. [34] proposed Mask R-CNN, a straightforward, adaptable, and versatile framework for person keypoint detection. Tao et al. [35] introduced a coarse-fine network for keypoint localization, utilizing multi-level supervisions to enhance accuracy. Kevin et al. [36] introduced a new aggregation method for precise keypoint predictions and innovative keypoint-based non-maximum-suppression and confidence scoring techniques. Sun et al. [37] introduced a pioneering network architecture named the cascaded pyramid network, comprising two stages: GlobalNet and RefineNet. Hu et al. [38] introduced advanced methods for precise human pose estimation, emphasizing low-level features and utilizing a cascaded network for sequential joint localization. Wei et al. [39] proposed straightforward and efficient baseline methods that served as valuable inspiration and evaluation tools for pioneering new concepts in the field. Lu et al. [40] proposed a Joint-candidate single person pose estimation method along with global maximum joints association to address detection challenges within crowded environments. In addition, Lu et al. [41] investigated the utilization of maximum likelihood estimation to devise efficient and effective regression-based techniques. Wang et al. [9] introduced a

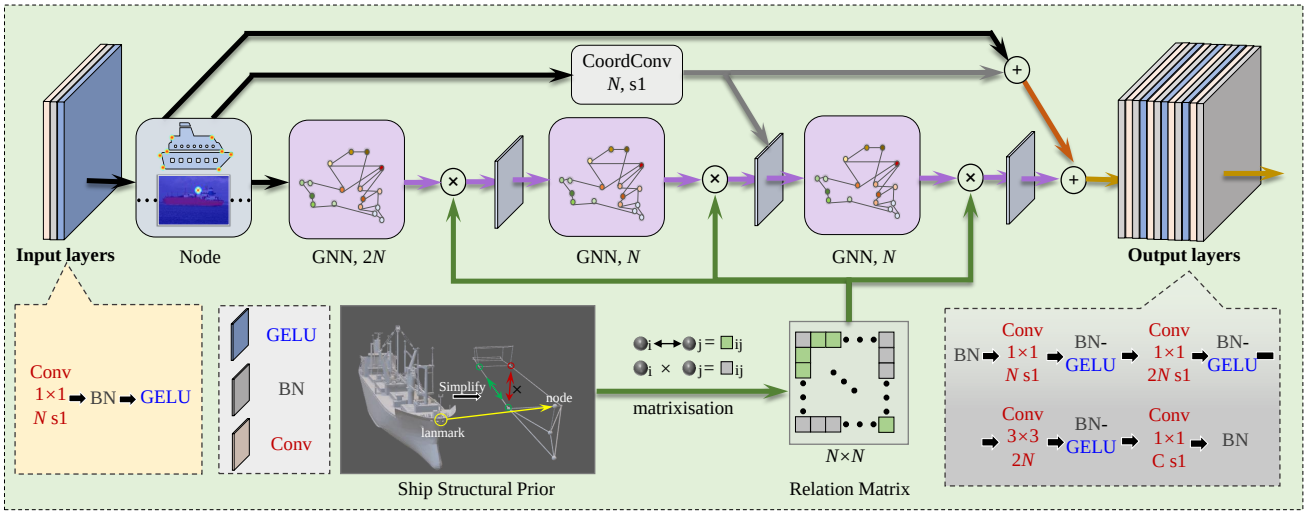


Fig. 2. Architecture of SPGG. First, the input of SPGG is transformed into N features, where N is the number of landmarks. Such features are then used graph ndoes in the GNNs. Next, guided by ship structural priors, a relationship matrix aids in capturing inter-node feature correlations. In addition, coordinate convolutions spatially encode input features, enhancing feature representation by fusing with graph features.

versatile CNN capable of preserving high-resolution representations throughout the entirety of the process. The Transformer [12] revolutionized natural language processing with self-attention, surpassing CNN limitations and extending its success to computer vision for diverse visual tasks. Hrformer [42] enhanced human pose estimation with local window self-attention, and ViTPose [10] improved performance using visual transformers for feature extraction. However, our experiments indicate that directly using Transformers for limited data landmark tasks underperforms compared to CNN-based methods, highlighting the need to optimise Transformer utilisation for these tasks. Given the challenges in collecting ship data and the high cost of manual annotation, top-down algorithms are preferable. Within this framework, heatmap prediction, which predicts N heatmaps representing the spatial distribution of landmarks, generally outperforms coordinate regression by focusing on local features, although it may overlook connections between keypoints.

III. Method

As shown in Fig. 1, SPG-GT consists of two important blocks, namely Structural Prior Guided GNN and Structural Prior Guided Transformers, referred to as SPGG and SPGT. It takes ship images as input, and first predicts a heatmap for each landmark through a simple encoder composed of a backbone network and a convolutional layer. Given a ship image with dimensions $[W, H, c]$, where W and H represent the width and the height of the image, and c refers to the number of channels in the image, the convolutional layer produces a feature map of size $[b, N, w, h]$, where b represents the batch size, N is the number of predicted landmark points ($N = 20$ in this work), and w and h are set to $1/4$ of W and H, respectively. Then, the feature map goes through 5 SPGG and 3 SPGT blocks to refine the learned heatmap, where the scores of real landmarks (i.e. inliers) are enhanced and those of fake landmarks (i.e. outliers) are suppressed. The framework is optimized by comparing the predicted heatmaps ($\hat{y}_i$) with the ground truth (y_i):

$$L_{\text{KMSE}} = \frac{1}{N}\sum_{i=1}^{N} \|y_i - \hat{y}_i\|_2^2 . \tag{1}$$

Next, we introduce SPGG and SPGT in detail.

A. SPGG Block

The structure of the SPGG block is illustrated in Fig. 2. The input data first go through a 3×3 convolution-batch normalization-GELU activation layer. Then, there are three branches. The first branch applies coordinate convolution [43] to encode spatial position features and obtain the feature $FM_{coor} = CoorConv(X_1)$. By spatially encoding the feature maps, the spatial information between different landmarks is enhanced. The second branch is the GNN branch. It represents the connectivity relationships in the form of a matrix, specifically an $N \times N$ matrix. The value in the i-th row and j-th column of the matrix represents the relationship between the i-th and the j-th landmarks. If there is a connection between them, the value is 1; otherwise, it is 0. The relational configuration stems from the prior of ship structures, with the physical interdependence between bow, stern, and specific landmarks of the superstructure determining their existence. Using this relational matrix, feature correlations are established between distinct landmarks of independent vessels.

To be specific, the input X_1 is flattened into N feature maps of size $w \times h$. Then, the relationship features extracted from the relational matrix through graph convolution are fused with the input features and expanded to twice the input feature size, followed by a ReLU activation. The feature size is then restored using graph convolution-activation layers and incorporated with spatial position features. The third branch is a residual structure that combines spatial position features, graph convolution features, and input features. These features are batch-normalized and enhanced with the embedding of

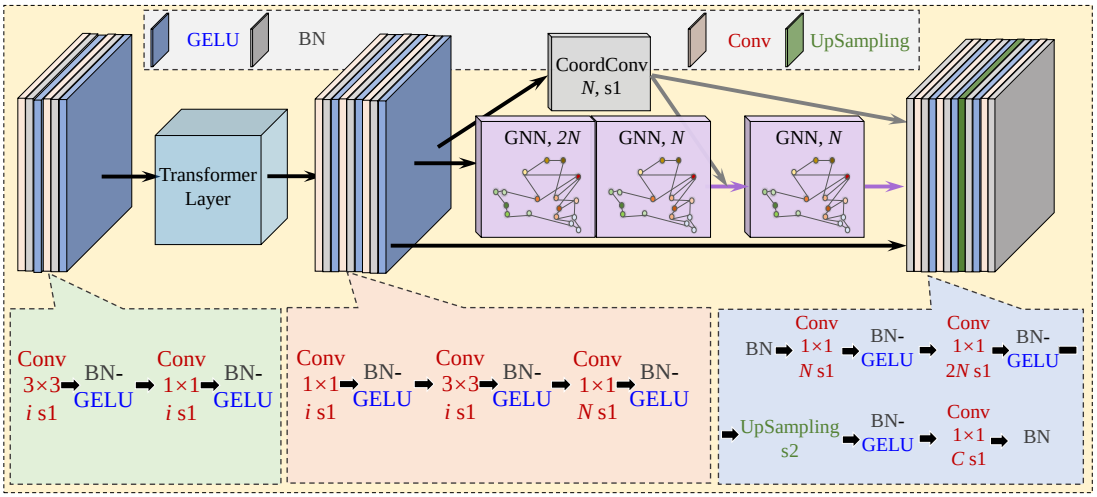


Fig. 3. Architecture of the SPGT block. BN represents batch normalization layer, GELU refers to the activation layer, Upsampling denotes the upsampling layer, s stands for stride, i indicates the number of input data features, and C represents the number of the output features.

spatial position encoding and connectivity relationship to strengthen the correlation of the predicted features. They are then directly input into a 3×3 convolution-BN-ReLU layer, which increases the dimensionality of the feature maps, matching the dimensionality of the input features. The feature maps are further processed through a 3×3 convolution-BN-ReLU module, where the stride is set to 1, unless otherwise specified. If no dimensionality increase is required, a regular 3×3 convolution-BN-ReLU is used. Finally, the feature maps are input into a 1×1 convolution-BN layer.

B. SPGT Block

The architecture of the SPGT block is illustrated in Fig. 3. The features are first extracted effectively through a 3×3 convolution-BN-GELU layer. Then, they are input into a 1×1 convolution-BN-GELU layer to enable information interaction between the predicted landmark feature maps. Next, a transformer layer is applied to establish global dependencies among the features using the self-attention mechanism. The feature representation is further enhanced using a 1×1 convolution-BN-GELU layer and a 3×3 convolution-BN-GELU layer. In all of the operations, the stride is set to 1 to maintain the size of the feature maps. Finally, the module employs an upsampling layer with a stride of 2. This contrasts with the stride of 1 in all other modules, effectively enlarging the size of the feature map. The introduction of features into the module serves to amplify graph connectivity relationships, as well as modify the dimensionality and the size of the feature maps.

C. Domain Transfer

Domain transfer refers to the directed shift of the data distribution from the source domain to the target domain. It aims to generate source domain data with the characteristic distribution of the target domain, which can address the challenges of collecting rare scenario data. Various algorithms can achieve this goal, but generating synthetic images provides a straightforward and visual way to observe the transformation of data distribution. Considering the significant differences in the number of samples among different classes in the current data target domain and the scarcity of target domain data, we choose CycleGAN [44] for domain transfer.

Specifically, CycleGAN consists of two generators and two discriminators. We adopt the network architecture proposed by Zhu et al. [44]. In the experimental setup, we use the training set of the SLAD dataset as the source domain, and different attributes, such as different daytime and weather conditions, as different target domains. The training was conducted with 200 epochs using the Adam optimizer [45] with a learning rate of 0.0002. We keep the same learning rate for the first 100 epochs and linearly decay it to zero over the next 100 epochs with a decay coefficient of 0.5.

IV. Experiment

We expand the SLAD++ dataset through domain adaptation based on the publicly available SLAD dataset. We compare SPG-GT with 8 state-of-the-art algorithms on the two datasets. Additionally, we compare 6 different backbone networks with 4 classic top-down frameworks. Furthermore, we carry out experimental validations to assess the impact of 13 types of domain shifts induced by external environments on ship landmark detection tasks.

A. Landmark Detection Datasets and Metrics

The Ship Landmark Dataset (SLAD) is the first publicly available dataset of ship images with landmark annotations, consisting of 12,199 images from 948 different ships. This dataset facilitates the analysis of deep features for landmark selection in ships. The dataset is divided into training, validation, and testing sets in a ratio of 7:1:2 using random sampling. The dataset provides instance-level bounding boxes and annotations for 20 different landmarks of the ships, covering a wide range of 121 common ship classes. The images in the dataset are collected under different weather and time conditions.

The Ship Landmark Dataset++ (SLAD++) integrates domain adaptation with SLAD to transform the data distribution

Fig. 4. The images display land and ocean scenes captured under snowy, cloudy, foggy, rainy, and sunny conditions. The land weather images on the left are sourced from [48], [49], while the ocean weather data on the right.

of the original training set to a distribution that includes various domain shifts related to maritime scenario. To ensure the smooth implementation of this work, we collect a collection of images prompted by the absence of visually annotated maritime datasets with weather indicators. This dataset consists of 25,518 images annotated with five weather conditions and eight time attributes, making it the first large-scale dataset of its kind. Ship large size complicate deductive data collection of real data, and as depicted in Fig. 4, there are discrepancies between maritime and common land-based data. Complex ship annotations hinder data sampling and algorithm development. The successful application of transfer learning from existing datasets is recognized. Sourced from ShipSpotting, the dataset contains ship details and exif data for weather and time annotations. Manual weather annotation is demanding, thus historical weather records were utilized to label capture times accurately. By leveraging GPS data—latitude, longitude, and UTC time—we minimized errors in dataset selection. Weather conditions were simplified into four main categories, with time periods determined by sunrise and sunset. Ship bounding boxes and semantic segmentation were accomplished through GroundingDino [46] and SAM [47] for object detection and segmentation.

In the context of landmark detection, inspired by the human pose estimation task, the similarity between the detected landmarks and the ground-truth landmarks is typically estimated using Object Keypoint Similarity (OKS). If the OKS value exceeds a certain threshold, the sample is considered as a positive detection, and vice versa. Consequently, we can calculate the Average Precision (AP) and Average Recall (AR) separately at various thresholds. Additionally, based on the size of the bounding box, the ships in SLAD are classified into three categories: small, medium, and large. We report AP and AR scores using different OKS thresholds, considering ships of different sizes. AP represents the average AP scores across 10 thresholds $(0.5, 0.55, \ldots, 0.9, 0.95)$ for all ships. $AP50$ and $AP75$ indicate the AP scores with thresholds of 0.5 and 0.75, respectively. AP_M and AP_L represent the AP scores for large and medium ships, respectively. The definitions of AR, $AR50$, $AR75$, AR_M, and AR_L scores are similar to those of the AP series.

In addition, we also report four metrics to measure the quality of generated images:

FID (Fréchet Inception Distance) [50]: It combines both statistical and perceptual features of images and utilizes the Inception network to extract features. A lower FID value indicates that the generated images are closer to real images. In Table V, FID is computed using synthetic data and target domain data, while FID2 is computed using synthetic data and source domain data. Unless otherwise specified, the remaining metrics are calculated using synthetic data and source domain data.

PSNR (Peak Signal-to-Noise Ratio) [51]: It evaluates the distortion level of an image by comparing the mean squared error between the original image and the reconstructed image. A higher PSNR value indicates better image quality.

SSIM (Structural Similarity Index) [52]: It assesses the distortion level of an image by comparing the structural similarity between the original image and the reconstructed image, taking into account factors such as brightness, contrast, and structure. The SSIM value ranges from 0 to 1, with values closer to 1 indicating better image quality.

LPIPS (Learned Perceptual Image Patch Similarity) [53]: It measures the similarity between images based on a perceptual learning network. Unlike traditional pixel-level or structural-level metrics, LPIPS considers human perception factors and aligns more with human judgment of image quality.

B. Compare with other methods and frameworks on SLAD

In this subsection, we evaluating 7 landmark detection methods on SLAD, including CPM [8], SimpleBaseline [39], RLE [41], ViTPose [10], HRNet [9], HRFormer [42], and Ours method.

- Convolutional Pose Machines (CPM [8]) uses cascaded convolutional neural networks to progressively extract and refine landmark information of the human body.
- Simple baselines for human pose estimation (SimpleBaseline [39]) provides a simple but effective baseline method for landmark detection. It employs a backbone network along with a few deconvolutional layers to predict heatmaps for the landmarks.
- Residual Log-likelihood EstimationRLE ([41]) is an effective approach for human pose regression that models the residual error between predicted and ground-truth joint coordinates.
- Simple Vision Transformer Baselines (ViTPose [10]) is an efficient plain vision Transformers for superior pose estimation performance..
- High-Resolution Net(HRNet [9]) applies a parallel multi-scale fusion strategy to integrate multi-resolution representations for higher accuracy and robustness in landmark detection.
- High-Resolution Transformer (HRFormer [42]) is a high-resolution transformer architecture that utilizes the feature fusion strategy in HRNet to capture multi-scale information and employs self-attention modules to model long-range dependencies.

In Tables I and III, we employ ResNet-50 as the backbone network for both SimpleBaseline and SimCC in accordance

TABLE I
AVERAGE PRECISION (AP) AND AVERAGE RECALL (AR) OF VARIOUS LANDMARK DETECTION METHODS ON SLAD.

Methods	Dataset	AP	AP50	AP75	AP_M	AP_L	AR	AR50	AR75	AR_M	AR_L	Time
CPM [8]	test	0.289	0.752	0.148	0.078	0.299	0.408	0.821	0.350	0.150	0.421	0.111
	val	0.302	0.796	0.156	0.103	0.311	0.415	0.853	0.354	0.178	0.426	0.115
SimpleBaseline [39]	test	0.518	0.908	0.523	0.172	0.533	0.613	0.922	0.659	0.265	0.630	0.080
	val	0.525	0.912	0.526	0.187	0.538	0.617	0.924	0.661	0.271	0.633	0.082
RLE [41]	test	0.471	0.857	0.456	0.129	0.486	0.544	0.880	0.568	0.190	0.560	0.067
	val	0.472	0.880	0.450	0.141	0.487	0.542	0.891	0.555	0.196	0.558	0.068
VitPose-Huge [10]	test	0.358	0.787	0.279	0.089	0.370	0.451	0.825	0.437	0.145	0.465	0.289
	val	0.351	0.796	0.260	0.118	0.361	0.442	0.836	0.410	0.200	0.453	0.291
VitPose-Large [10]	test	0.324	0.762	0.217	0.092	0.335	0.415	0.805	0.377	0.137	0.428	0.299
	val	0.315	0.759	0.214	0.119	0.323	0.401	0.801	0.354	0.184	0.411	0.297
VitPose-Base [10]	test	0.344	0.784	0.236	0.101	0.354	0.435	0.822	0.393	0.168	0.448	0.140
	val	0.331	0.784	0.230	0.110	0.341	0.422	0.821	0.379	0.165	0.433	0.142
VitPose-Small [10]	test	0.327	0.767	0.222	0.087	0.337	0.418	0.811	0.376	0.153	0.430	0.095
	val	0.320	0.765	0.221	0.100	0.330	0.411	0.815	0.357	0.180	0.422	0.096
HRFormer-Base [42]	test	0.597	0.933	0.645	0.192	0.615	0.686	0.945	0.749	0.290	0.705	0.301
	val	0.608	0.935	0.661	0.239	0.624	0.692	0.946	0.756	0.318	0.710	0.305
HRFormer-Small [42]	test	0.559	0.920	0.593	0.194	0.573	0.649	0.931	0.712	0.298	0.665	0.126
	val	0.571	0.933	0.615	0.230	0.585	0.654	0.943	0.718	0.316	0.670	0.133
Swin-Large [16]	test	0.253	0.668	0.118	0.062	0.261	0.345	0.739	0.269	0.098	0.357	0.254
	val	0.253	0.675	0.131	0.087	0.260	0.343	0.749	0.276	0.122	0.353	0.254
Swin-Base [16]	test	0.424	0.845	0.376	0.097	0.440	0.521	0.880	0.542	0.175	0.537	0.150
	val	0.421	0.858	0.354	0.136	0.433	0.515	0.889	0.516	0.224	0.529	0.151
Swin-Tiny [16]	test	0.421	0.845	0.373	0.127	0.434	0.516	0.880	0.539	0.204	0.531	0.076
	val	0.422	0.857	0.338	0.171	0.434	0.515	0.889	0.496	0.259	0.527	0.081
HRNet-W32 [9]	test	0.566	0.922	0.604	0.200	0.579	0.652	0.939	0.716	0.321	0.668	0.078
	val	0.569	0.934	0.596	0.251	0.583	0.653	0.946	0.708	0.333	0.668	0.081
HRNet-W48 [9]	test	0.574	0.924	0.626	0.181	0.591	0.666	0.940	0.734	0.293	0.684	0.212
	val	0.592	0.936	0.647	0.251	0.606	0.672	0.946	0.737	0.322	0.688	0.213
Ours	test	0.585	0.931	0.637	0.191	0.601	0.671	0.942	0.736	0.295	0.689	0.133
	val	0.594	0.936	0.655	0.228	0.609	0.675	0.947	0.748	0.306	0.692	0.146

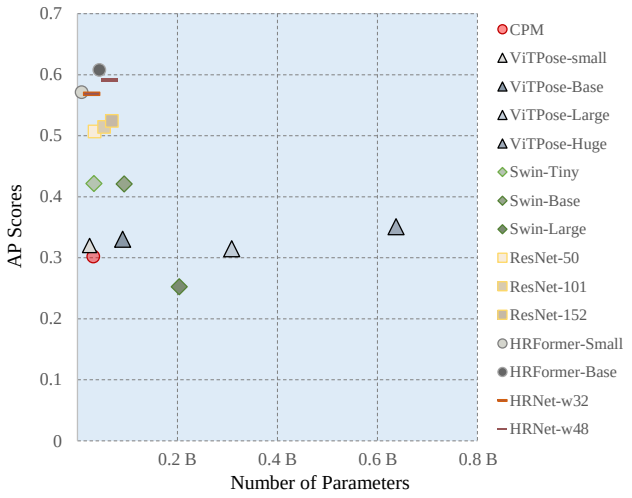


Fig. 5. Performance of various CNN-based and Transformer-based landmark detection methoeds on the SLAD dataset.

with the official implementation. Similarly, CPM and RLE frameworks also adopt the original network architectures as their backbone networks. For ViTPose, we opt for the backbone network of its ViTPose-Base version. For HRNet, we report the performance of its two variants, i.e. HRNet-W32 and HRNet-W48, where the suffix indicates the width (or number of channels) of high-resolution sub-networks in the last three stages. The details of the implementation of HRFormer can be found in [42]. Besides, it is worth noting that all the methods are conducted under one setting of input size, i.e. 256×192. To train the above methods on SLAD, we apply the Adam optimizer with a learning rate of 0.0005, betas (0.9, 0.999), and weight decay 0.01. The batch size is set to 64 for most of the experiments except for Swin-Large, ViTPose-Large, and ViTPose-Huge. All the experiments were performed on a Ubuntu 22.04 system with 4 NVIDIA RTX 4090 GPUs. The methods implemented above are based on MMPose [54]. The time unit is seconds per iteration, and the testing batch size is 32.

Table I lists the evaluation results, from which we obtain

TABLE II
AVERAGE PRECISION (AP) AND AVERAGE RECALL (AR) OF VARIOUS LANDMARK DETECTION METHODS ON SLAD.

Methods	Dataset	AP	AP50	AP75	AP_M	AP_L	AR	AR50	AR75	AR_M	AR_L	Time
ResNet-50 + DeepPose	test	0.160	0.533	0.051	0.015	0.172	0.306	0.723	0.213	0.071	0.317	0.042
	val	0.167	0.534	0.045	0.014	0.178	0.303	0.711	0.199	0.053	0.315	0.045
ResNet-50 + RLE	test	0.407	0.831	0.361	0.089	0.422	0.486	0.850	0.485	0.133	0.502	0.039
	val	0.407	0.824	0.340	0.085	0.420	0.482	0.847	0.468	0.131	0.498	0.044
ResNet-50 + SimpleBaseline	test	0.504	0.896	0.511	0.155	0.519	0.595	0.917	0.641	0.242	0.612	0.051
	val	0.507	0.911	0.510	0.196	0.519	0.597	0.925	0.635	0.280	0.611	0.055
ResNet-50 + SPG-GT	test	0.507	0.895	0.513	0.141	0.523	0.601	0.915	0.649	0.241	0.618	0.083
	val	0.512	0.911	0.501	0.172	0.527	0.603	0.928	0.636	0.259	0.619	0.089
ResNet-101 + DeepPose	test	0.169	0.531	0.052	0.024	0.178	0.312	0.724	0.222	0.091	0.323	0.051
	val	0.170	0.533	0.053	0.041	0.179	0.307	0.717	0.220	0.094	0.317	0.056
ResNet-101 + RLE	test	0.448	0.845	0.430	0.127	0.462	0.521	0.866	0.541	0.178	0.537	0.050
	val	0.451	0.857	0.410	0.125	0.465	0.522	0.879	0.528	0.196	0.537	0.057
ResNet-101 + SimpleBaseline	test	0.509	0.897	0.520	0.148	0.524	0.600	0.913	0.650	0.240	0.617	0.064
	val	0.514	0.898	0.523	0.176	0.528	0.601	0.920	0.643	0.263	0.616	0.071
ResNet-101 + SPG-GT	test	0.506	0.898	0.504	0.183	0.519	0.596	0.915	0.640	0.268	0.612	0.096
	val	0.519	0.914	0.512	0.179	0.532	0.600	0.924	0.636	0.251	0.617	0.110
ResNet-152 + DeepPose	test	0.184	0.545	0.064	0.023	0.195	0.333	0.739	0.247	0.108	0.343	0.064
	val	0.189	0.592	0.068	0.027	0.199	0.328	0.754	0.244	0.084	0.339	0.070
ResNet-152 + RLE	test	0.471	0.857	0.456	0.129	0.486	0.544	0.880	0.568	0.190	0.560	0.067
	val	0.472	0.880	0.450	0.141	0.487	0.542	0.891	0.555	0.196	0.558	0.068
ResNet-152 + SimpleBaseline	test	0.518	0.908	0.523	0.172	0.533	0.613	0.922	0.659	0.265	0.630	0.080
	val	0.525	0.912	0.526	0.187	0.538	0.617	0.924	0.661	0.271	0.633	0.082
ResNet-152 + SPG-GT	test	0.517	0.909	0.523	0.154	0.532	0.611	0.922	0.659	0.240	0.628	0.107
	val	0.531	0.911	0.554	0.188	0.545	0.614	0.921	0.664	0.257	0.631	0.117
HRNet-w32	test	0.566	0.922	0.604	0.200	0.579	0.652	0.939	0.716	0.321	0.668	0.078
	val	0.569	0.934	0.596	0.251	0.583	0.653	0.946	0.708	0.333	0.668	0.081
HRNet-w32 + SPG-GT	test	0.559	0.931	0.589	0.199	0.572	0.646	0.942	0.703	0.305	0.663	0.121
	val	0.574	0.946	0.596	0.207	0.589	0.655	0.951	0.709	0.271	0.673	0.119
HRNet-w48	test	0.574	0.924	0.626	0.181	0.591	0.666	0.940	0.734	0.293	0.684	0.212
	val	0.592	0.936	0.647	0.251	0.606	0.672	0.946	0.737	0.322	0.688	0.213
HRNet-w48 + SPG-GT	test	0.585	0.931	0.637	0.191	0.601	0.671	0.942	0.736	0.295	0.689	0.133
	val	0.594	0.936	0.655	0.228	0.609	0.675	0.947	0.748	0.306	0.692	0.146

the following observations:

(1) CPM produces the worst performance in ship landmark detection, except for directly using a transformer as the backbone, with the AP score being at least 17% lower than other CNN-based methods. The primary reason is that CPM is the sole two-stage method among the mentioned approaches in the table. This particular approach carries out feature extraction and landmark estimation in distinct stages, potentially resulting in the extracted features from the backbone network not being entirely optimal for meeting the demands of the landmark detection task.

(2) With the training same dataset, Transformer-based ViTPose using ViT-base as the backbone network for feature extraction achieves an AP score 27.7% lower than HRFormer on validation set. This percentage decreases to 25.3% on the test set. The training difficulty of Transformers, which is higher than that of CNNs with equivalent parameters, makes it challenging to achieve the expected performance when directly using a Transformer as a backbone network.

(3) Both SimpleBaseline and RLE are top-down methods. With the same training dataset, SimpleBaseline ResNet-152 as the backbone network performed better than RLE. Which directly regresses coordinates, and has an advantage in terms of speed. This also indicates that different prediction methods have an impact on the prediction results. Under the assumption of a fixed backbone network, modifying the prediction head affects the network's prediction performance.

We also conducted comparative experiments of the baselines with transformer-based backbones, including Swin-Tiny, Swin-Base, and Swin-Large [16], HRFormer-Small and HRFormer-Base [42], ViTPose-Small, ViTPose-Base, ViTPose-Large, and ViTPose-Huge [10]. In addition, we included CNN-based backbones such as ResNet-50, ResNet-101, ResNet-152, HRNet-w32, and HRNet-w48.

By comparing these networks from Fig. 5, we observed that CNN-based backbones can improve network performance by

Fig. 6. Prediction results produced by HRNet-w48 and SPG-GT trained on the same dataset with consistent training hyperparameters.

simply increasing depth (ResNet series) or width (HRNet). However, when using transformer-based backbones, simply increasing the network's size does not necessarily lead to performance improvement. Under the same parameter settings and training on the SLAD dataset, CNN-based backbone networks outperform pure transformer-based networks. We can also observe that HRFormer-base has slightly fewer parameters compared to HRNet-w48, but achieves slightly higher AP scores. However, as shown in Table I, HRFormer-base has slower computational speed than HRNet-w48, indicating a higher computational load, which is also evident during the training process.

We compared the SPG-GT with the previously mentioned SimpleBaseline and RLE, two top-down detection frameworks. Additionally, we also compared it with the detection framework of Deep Pose, which is the first work to apply deep learning to human pose estimation [55]. In terms of experimental design, we used three different depth networks, namely ResNet-50, ResNet-101, and ResNet-152, as backbone networks to compare the differences among the four detection frameworks. We also compared the SPG-GT and the original network, both using HRNet as the backbone network. The experimental parameter settings were the same as the aforementioned experiments.

From Table II, we can see that when using the same backbone, SPG-GT performs better than the competing methods for landmark detection, but it requires longer time for inference. Despite having a smaller number of parameters than ResNet-50, HRNet-w32 has longer inference time, indicating that it has higher computational complexity than ResNet-50. Similarly, HRNet-w48 outperforms ResNet-152, despite having a larger number of parameters, suggesting that its full-scale feature inference approach is more suitable for landmark detection. Thus, HRNet is chosen as the backbone network for SPG-GT. Fig. 6 shows that compared with the original HRNet, SPG-GT effectively reduces the weights of erroneously identified regions and enhance the weights of correctly identified regions with low confidence.

C. Compare with other methods and frameworks on SLAD++

In this subsection, we evaluating 7 landmark detection methods on SLAD, including CPM [8], SimpleBaseline [39], RLE [41], ViTPose [10], HRNet [9], HRFormer [42], and Ours method. The methods and training environment details involved in these experiments are consistent with the previous section.

Table I lists the evaluation results, from which we obtain the following observations:

(1) Compared to training on SLAD, all methods show a significant improvement in performance after training on SLAD++. On the SLAD test set, all methods exhibit an average AP improvement of 8.4%, with an average improvement of 9.2% on the validation set. Particularly noteworthy is that the performance improvement varies across different methods; CPM, RLE, ViTPose, and Swin show performance enhancements on the test or validation sets close to or exceeding 10%. Notably, Swim-Large even achieves a 28% performance boost on the validation set. In contrast, HRNet, HRFormer, and our method show improvements below 5%, yet their overall performance surpasses that of other methods. Enhancing the quality of training data plays a crucial role in achieving ship landmark detection.

(2) For the 4 different scale models of VitPose, ranging from VitPose-Small to VitPose-Base and VitPose-Large, there is a trend where larger parameters lead to better performance. However, the increase in parameters from VitPose-Large to VitPose-Huge does not yield a corresponding performance improvement. This observation is also evident in Swin-Tiny, Swin-Base, and Swin-Large models. In other words, simply scaling up the network parameters is not the optimal solution for ship landmark detection tasks.

(3) Both SimpleBaseline and RLE are top-down methods. When utilizing ResNet-152 as the backbone network, during training on SLAD, SimpleBaseline outperforms RLE in terms of AP performance, whereas when trained on SLAD++, the performance of RLE is comparable to or slightly better than SimpleBaseline. This suggests that the training complexity of the RLE method is higher than that of SimpleBaseline; regressing coordinates directly poses greater training challenges compared to predicting heatmaps. Heatmap prediction methods are more suitable when dealing with limited data.

By comparing the results in Table IV, we can see that when using the same backbone trained on SLAD++, the inference speed remains nearly consistent with that trained on SLAD.

Deep Pose and RLE have similar inference times, with

TABLE III
AVERAGE PRECISION (AP) AND AVERAGE RECALL (AR) OF VARIOUS LANDMARK DETECTION METHODS ON SLAD++.

Methods	Dataset	AP	AP50	AP75	AP_M	AP_L	AR	AR50	AR75	AR_M	AR_L	Time
CPM [8]	test	0.409	0.867	0.329	0.113	0.422	0.522	0.894	0.537	0.196	0.537	0.112
	val	0.425	0.889	0.341	0.193	0.435	0.533	0.910	0.540	0.263	0.545	0.116
SimpleBaseline [39]	test	0.550	0.910	0.575	0.178	0.566	0.638	0.927	0.686	0.274	0.655	0.079
	val	0.559	0.922	0.581	0.182	0.575	0.640	0.932	0.682	0.284	0.657	0.084
RLE [41]	test	0.560	0.871	0.592	0.131	0.579	0.627	0.890	0.669	0.196	0.647	0.065
	val	0.562	0.893	0.608	0.133	0.579	0.627	0.906	0.674	0.202	0.647	0.068
VitPose-Huge [10]	test	0.480	0.869	0.470	0.136	0.494	0.584	0.892	0.613	0.221	0.601	0.289
	val	0.475	0.879	0.470	0.170	0.488	0.581	0.900	0.620	0.271	0.596	0.290
VitPose-Large [10]	test	0.480	0.861	0.472	0.123	0.495	0.575	0.886	0.607	0.217	0.591	0.299
	val	0.477	0.858	0.471	0.145	0.491	0.568	0.889	0.593	0.235	0.584	0.301
VitPose-Base [10]	test	0.457	0.854	0.435	0.123	0.470	0.543	0.876	0.562	0.205	0.559	0.137
	val	0.457	0.846	0.442	0.137	0.470	0.541	0.879	0.552	0.239	0.555	0.142
VitPose-Small [10]	test	0.419	0.825	0.372	0.090	0.433	0.518	0.859	0.522	0.176	0.533	0.093
	val	0.425	0.834	0.377	0.126	0.438	0.522	0.864	0.523	0.222	0.536	0.095
HRFormer-Base [42]	test	0.618	0.931	0.680	0.230	0.634	0.704	0.947	0.768	0.325	0.722	0.302
	val	0.626	0.945	0.684	0.232	0.643	0.707	0.957	0.763	0.308	0.726	0.302
HRFormer-Small [42]	test	0.598	0.921	0.656	0.213	0.614	0.686	0.938	0.746	0.314	0.703	0.119
	val	0.605	0.946	0.651	0.254	0.620	0.689	0.951	0.742	0.335	0.705	0.130
Swin-Large [16]	test	0.525	0.897	0.546	0.136	0.542	0.624	0.918	0.676	0.230	0.643	0.254
	val	0.533	0.907	0.552	0.189	0.547	0.628	0.924	0.675	0.294	0.643	0.258
Swin-Base [16]	test	0.522	0.907	0.538	0.145	0.537	0.620	0.921	0.667	0.237	0.637	0.152
	val	0.518	0.907	0.527	0.195	0.531	0.614	0.924	0.656	0.269	0.630	0.150
Swin-Tiny [16]	test	0.530	0.899	0.554	0.128	0.546	0.623	0.918	0.671	0.226	0.641	0.076
	val	0.527	0.908	0.526	0.192	0.539	0.619	0.925	0.656	0.294	0.635	0.077
HRNet-W32 [9]	test	0.599	0.932	0.658	0.200	0.617	0.688	0.940	0.754	0.306	0.706	0.083
	val	0.610	0.934	0.671	0.243	0.627	0.694	0.945	0.759	0.339	0.711	0.078
HRNet-W48 [9]	test	0.598	0.933	0.651	0.228	0.612	0.683	0.943	0.744	0.338	0.700	0.101
	val	0.612	0.935	0.688	0.260	0.625	0.689	0.949	0.763	0.359	0.704	0.104
Ours	test	0.605	0.930	0.667	0.225	0.620	0.689	0.946	0.753	0.325	0.707	0.134
	val	0.620	0.937	0.685	0.283	0.634	0.699	0.949	0.767	0.376	0.714	0.136

RLE being better than Deep Pose. SimpleBaseline outperforms the other two methods, although its inference time is slightly higher. SPG-GT performs better than the other three methods in landmark detection, but it requires more time for inference. Compared to SimpleBaseline, Compared to SimpleBaseline using ResNet-50, ResNet-101, and ResNet-152, SPG-GT shows improvements in average precision (AP) is improved by -0.25%, 1%, and 0.45%, respectively.

For HRNet-w32 and HRNet-w48, using SPG-GT increases the parameter count by 1.83% and 0.82%, respectively, and improves the AP by 0.2% and 0.75%, respectively. Despite having a smaller parameter count than ResNet-50, HRNet-w32 has a longer inference time, indicating that it has higher computational complexity than ResNet-50. Similarly, HRNet-w48 outperforms ResNet-152, despite having a larger parameter count, suggesting that its full-scale feature inference approach is more suitable for landmark detection. Ultimately, HRNet is chosen as the backbone network for the model.

An important distinction lies in the RLE framework, which shows significant overall performance improvement after training on SLAD++. It slightly outperforms the SimpleBaseline framework on ResNet-50, 101, and 152, indicating that the RLE framework demands higher quality training data and presents greater training complexity.

Comparing SPG-GT and SimpleBaseline with ResNet-50 as the backbone, SimpleBaseline exhibits a 0.01% higher AP performance on the validation set and a -0.5% improvement on the test set, with faster inference speed. However, with ResNet-101 and 152 as the backbone networks, SPG-GT surpasses SimpleBaseline by 0.8% and 0.5% on the validation set, and by 1.2% and 0.4% on the test set, respectively.

For HRNet-w32 and HRNet-w48, using SPG-GT also leads to performance enhancements. However, in terms of computational speed, due to the consistent heatmap output size from different backbones, the additional computation time of SPG-GT remains approximately the same. Experimental results indicate that SPG-GT demonstrates stable performance on both SLAD and SLAD++, with lower training complexity. Particularly, it performs best when utilizing HRNet-W48 as the backbone network, surpassing the original model on both

TABLE IV
AVERAGE PRECISION (AP) AND AVERAGE RECALL (AR) OF VARIOUS LANDMARK DETECTION METHODS ON SLAD++.

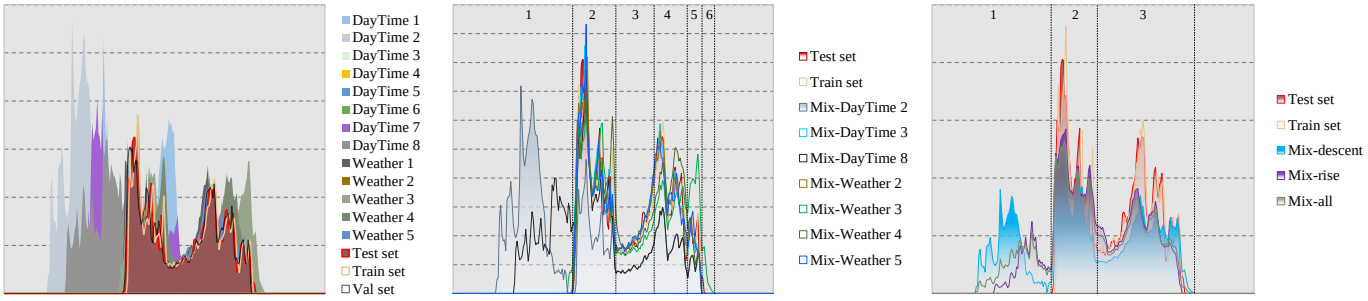
Methods	Dataset	AP	AP50	AP75	AP_M	AP_L	AR	AR50	AR75	AR_M	AR_L	Time
ResNet-50 + DeepPose	test	0.313	0.702	0.230	0.033	0.334	0.481	0.839	0.476	0.128	0.497	0.038
	val	0.324	0.728	0.243	0.075	0.340	0.481	0.842	0.475	0.163	0.496	0.042
ResNet-50 + RLE	test	0.537	0.898	0.550	0.166	0.553	0.637	0.919	0.679	0.259	0.655	0.052
	val	0.544	0.911	0.563	0.157	0.560	0.632	0.928	0.674	0.253	0.650	0.059
ResNet-50 + SimpleBaseline	test	0.545	0.882	0.565	0.145	0.563	0.614	0.898	0.652	0.209	0.632	0.039
	val	0.552	0.902	0.572	0.186	0.567	0.615	0.911	0.647	0.255	0.632	0.042
ResNet-50 + SPG-GT	test	0.540	0.906	0.561	0.162	0.556	0.633	0.922	0.680	0.248	0.650	0.085
	val	0.552	0.913	0.584	0.184	0.569	0.638	0.929	0.684	0.257	0.656	0.090
ResNet-101 + DeepPose	test	0.326	0.717	0.253	0.036	0.346	0.491	0.848	0.496	0.132	0.508	0.053
	val	0.333	0.749	0.258	0.075	0.352	0.489	0.854	0.495	0.155	0.504	0.056
ResNet-101 + RLE	test	0.563	0.878	0.590	0.173	0.579	0.626	0.896	0.661	0.249	0.643	0.054
	val	0.556	0.892	0.571	0.161	0.572	0.617	0.906	0.642	0.241	0.635	0.056
ResNet-101 + SimpleBaseline	test	0.544	0.911	0.579	0.157	0.560	0.633	0.925	0.686	0.249	0.651	0.064
	val	0.549	0.913	0.571	0.170	0.564	0.635	0.928	0.683	0.261	0.653	0.069
ResNet-101 + SPG-GT	test	0.556	0.910	0.569	0.194	0.572	0.639	0.926	0.676	0.282	0.656	0.108
	val	0.557	0.910	0.570	0.197	0.572	0.639	0.925	0.676	0.284	0.656	0.106
ResNet-152 + DeepPose	test	0.343	0.731	0.280	0.043	0.363	0.511	0.856	0.523	0.158	0.527	0.065
	val	0.359	0.765	0.295	0.090	0.377	0.511	0.864	0.526	0.190	0.526	0.069
ResNet-152 + RLE	test	0.560	0.871	0.592	0.131	0.579	0.627	0.890	0.669	0.196	0.647	0.065
	val	0.562	0.893	0.608	0.133	0.579	0.627	0.906	0.674	0.202	0.647	0.068
ResNet-152 + SimpleBaseline	test	0.550	0.910	0.575	0.178	0.566	0.638	0.927	0.686	0.274	0.655	0.079
	val	0.559	0.922	0.581	0.182	0.575	0.640	0.932	0.682	0.284	0.657	0.084
ResNet-152 + SPG-GT	test	0.554	0.900	0.587	0.177	0.570	0.641	0.919	0.690	0.273	0.659	0.106
	val	0.564	0.923	0.587	0.179	0.579	0.645	0.931	0.688	0.276	0.663	0.112
HRNet-w32	test	0.599	0.932	0.658	0.200	0.617	0.688	0.940	0.754	0.306	0.706	0.083
	val	0.610	0.934	0.671	0.243	0.627	0.694	0.945	0.759	0.339	0.711	0.078
HRNet-w32 + SPG-GT	test	0.597	0.932	0.646	0.227	0.613	0.688	0.947	0.749	0.322	0.705	0.115
	val	0.616	0.945	0.686	0.242	0.631	0.696	0.950	0.760	0.320	0.714	0.127
HRNet-w48	test	0.598	0.933	0.651	0.228	0.612	0.683	0.943	0.744	0.338	0.700	0.101
	val	0.612	0.935	0.688	0.260	0.625	0.689	0.949	0.763	0.359	0.704	0.104
HRNet-w48 + SPG-GT	test	0.605	0.930	0.667	0.225	0.620	0.689	0.946	0.753	0.325	0.707	0.134
	val	0.620	0.937	0.685	0.283	0.634	0.699	0.949	0.767	0.376	0.714	0.136

validation and test sets. While HRFormer-Base shows a slight improvement of 0.6% and 1.3% on the validation and test sets compared to SPG-GT, its computational speed is 223.7% slower than SPG-GT.

D. Comparative analysis of domain transfer

In this experiment, to evaluate the impact of different domain distributions on landmark detection tasks, we first used the training set of SLAD as the source domain to prevent the influence of data distribution from the validation set or test set. The target domain consisted of 8 daytime attributes and 5 weather attributes. Additionally, we assumed that the images represents samples collected from the real world and its distribution reflects the difficulty of obtaining different data, as shown in Table V. Based on this assumption, we conducted experiments on domain transfer using target domain samples with varying difficulty levels and quantities. Table V presents four metrics for different attribute domains. From the results, we can observe that as the number of target domain samples increases, the transfer effect of the generative network improves. This is evident through higher SSIM and lower LPIPS, indicating better preservation of the structural characteristics of the source domain data. Moreover, the domain distribution transfer also improves in terms of quality, as reflected by lower FID and FID2, and higher PSNR, indicating better image quality.

Based on this, we divided the synthetic data into multiple categories. The first category consists of target domain samples with larger quantities (more than 3000), which have a reliable distribution and good image quality. Specifically, their FID is less than 10, PSNR is greater than 30, SSIM is greater than 0.9, and LPIPS is less than 0.1. This category includes Daytime 3, 4, 5, and 6, as well as Weather 1, 2, and 5. The other categories have limited numbers of target domain samples, leading to less reliable distributions and lower quality of generated images. These samples have a PSNR lower than



(a) Synthetic images and SLAD (b) Mixed training sets with single-domain (c) Mixed training sets with multi-domain

Fig. 7. The figure depicts the average pixel distributions of the dataset ensemble, which we refer to as the domain distribution map. Subfigure (a) displays the average pixel distribution of multiple domain transfer data sets along with the training and testing sets of SLAD. Subfigure (b) showcases the blended domain distribution of a subset of individual domain transfer data sets mixed with the training set, in addition to the domain distribution of SLAD's training and testing sets. Subfigure (c) demonstrates the blended domain distribution of three different domain transfer data sets mixed with the training set, as well as the domain distribution of SLAD's training and testing sets.

TABLE V
QUANTITATIVE MEASUREMENTS OF DOMAIN TRANSFER DATA INCLUDE EIGHT TYPES OF DAYTIME AND FIVE TYPES OF WEATHER, ENCOMPASSING FIVE EVALUATION METRICS: FID, FID2, PSNR, SSIM, AND LPIPS.

Name	FID↓	FID2↓	PSNR↑	SSIM↑	LPIPS↓	Targets
DayTime 1	47.132	23.233	27.981	0.727	0.232	215
DayTime 2	139.722	94.556	27.971	0.418	0.437	41
DayTime 3	8.832	1.563	30.538	0.970	0.021	3566
DayTime 4	6.688	1.615	30.985	0.974	0.018	6447
DayTime 5	7.409	1.584	30.825	0.974	0.018	6700
DayTime 6	2.243	1.811	30.947	0.973	0.019	6921
DayTime 7	50.867	16.262	27.833	0.755	0.207	188
DayTime 8	82.891	49.249	27.896	0.618	0.331	101
Weather 1	6.912	1.412	30.919	0.974	0.019	6168
Weather 2	9.343	2.382	30.468	0.968	0.025	4125
Weather 3	58.458	19.019	28.257	0.774	0.212	75
Weather 4	58.311	12.491	28.386	0.837	0.175	89
Weather 5	7.694	1.977	30.587	0.970	0.024	3235

30, and further subdivisions can be made based on the metrics in the table. The second category represents a reasonably reliable distribution transfer, with FID fluctuating around 50, FID2 below 20, SSIM above 0.7 but less than 0.9, and LPIPS between 0.1 and 0.3. This category includes Daytime 1, 7, and Weather 3, 4. The third category represents a lower reliability in distribution transfer, with FID greater than 80 but less than 100, FID2 greater than 20, SSIM less than 0.7, and LPIPS greater than 0.3. Specifically, this includes Daytime 8. The fourth category represents extremely low reliability in distribution transfer and poorer generated image quality, specifically referring to Daytime 2.

To visually demonstrate the changes in domain transfer, we synthesized a corresponding sample for each domain for every training set sample. Then, we obtained an average sample for each set by calculating the mean, as follows: $I_{mean} = \frac{1}{n}\sum_{i=1}^{n} I_i$represents the sample resized to (256, 256) within the set. Fig. 7(a) shows the distribution of the average samples for each domain. From the Fig. 7(a), we can observe that the training set and test set are randomly sampled from the same batch of data, and the samples within each set exhibit independent and identically distributed characteristics. Daytime 2, 3, and 8 exhibit significant distribution discrepancies, which correspond to the FID2 scores. This also indicates that, even with fixed spatial positions in the target space, differences in data acquisition conditions can have a significant impact on the data.

To visually demonstrate the impact of domain transfer on image features, as shown in Fig. 8, we selected four tasks for intuitive presentation, including a classification task that focuses on local key features, detection and semantic segmentation tasks that focus on overall target features, and a landmark detection task that focuses on specific regions of the target.

Classification, we employing the Contrastive Language-Image Pretraining (CLIP, [56]), with ResNet-50 as the visual encoder backbone. The text "the ship" was used, and the CAM (Class Activation Mapping) images of the visual encoder were visualized by Grad-CAM [57]. As shown in the Fig. 8, with different domain transfers of the same sample, the CAM images exhibit changes in the focused regions.

Detection, we employing GroundingDino [46], also using "the ship" as the prompt. The green boxes represent the ground truth detection bounding boxes, while the red boxes represent the detection boxes for the current sample. From the Fig. 8, we can see that both weather and daytime changes can potentially affect the results of object detection.

Semantic segmentation, we employing Segment Anything (SAM, [47]). The green part represents the input of the green bounding boxes from the previous detection task, and the red part represents the input of the red boxes. The orange part represents the overlapping region between the two inputs. Apart from the segmentation changes caused by the variations in bounding boxes, there are still corresponding influences, such as Weather 1.

landmark detection, we employing the trained the HRNet-

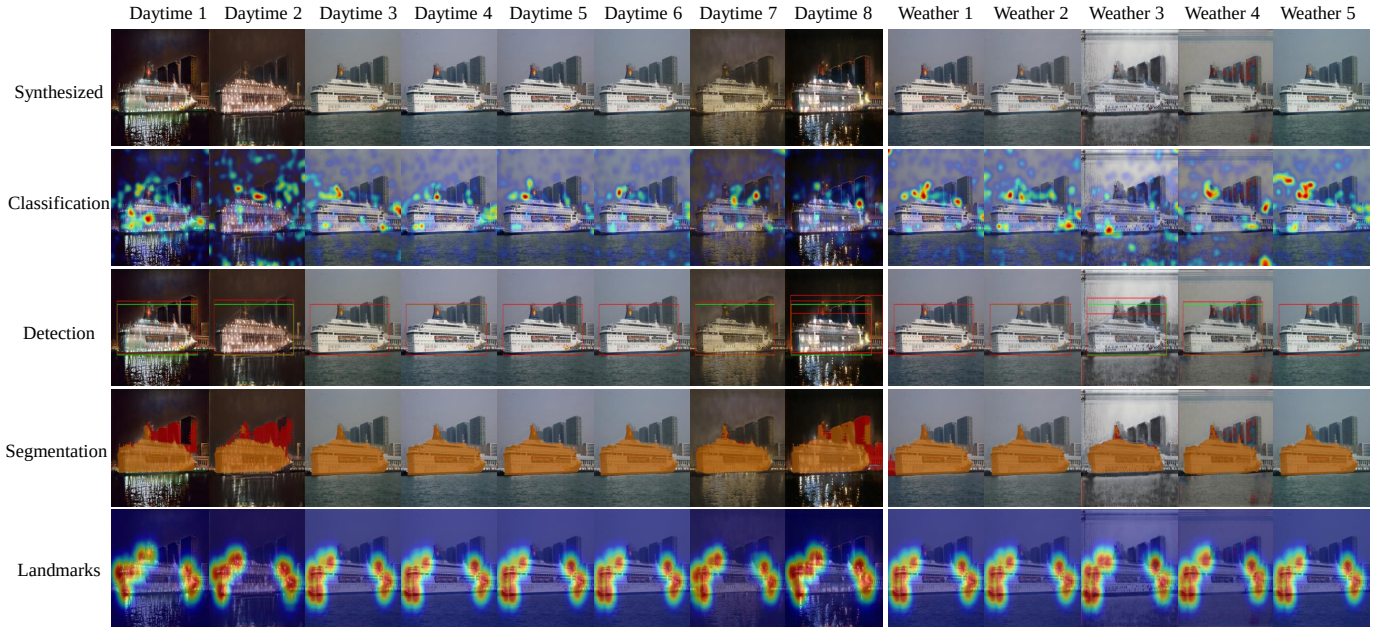


Fig. 8. The figure shows the synthesized images after domain transfer and visualizes the impact of domain variation on different tasks. From top to bottom, the figure shows the synthesized images, the CAM image variation for classification tasks, the bounding box variation for object detection tasks, the mask variation for semantic segmentation tasks, and the landmark prediction variation for landmark detection tasks.

w32 model on SLAD. Taking Daytime 2 and 3 as examples, domain changes have two effects on landmark predictions. One effect is related to the number of landmarks predicted. Weather 2 detects three landmarks in the rear of the ship, while Weather 3 detects four landmarks. The second effect is the change in the accuracy of the predicted points. In Weather 2, the two points predicted above the rear of the ship are significantly higher in depth compared to Weather 3, indicating higher numerical values.

E. Domain Transfer on landmark detection Experiment

The previous experiments demonstrated that SPG-GT outperforms other frameworks, especially SimpleBaseline, when trained on the SLAD dataset. In this experiment, we chose SPG-GT and introduced domain transfer to the training data to test its impact on the results. We used two different versions of the backbone network, HRNet-w32 and HRNet-w48. The parameter settings during training remained the same as before, but we used the formula to perform mixed training with domain transfer. We analyzed the performance metrics on the test set and reported the experimental results of incorporating thirteen domain transfer data into single-domain style transfer training in Table VI. After introducing an equal amount of synthesized domain transfer data, the trained networks showed performance improvements on the test set. However, the extent of improvement varied across different categories. For example, Daytime 2 only improved by 0.55%, which is much lower than the improvements in other categories. This category only had 41 target domain samples, and the generated test metrics were also much lower than the others. Next, we will explore whether the impact of domain transfer on landmark detection is solely influenced by the quantity and quality of the generated samples.

We observed that Weather 2, 3, 4, and 5 showed improvements of 2%, 1.35%, 1.8%, and 2.7% respectively. However, referring to Table V, it can be seen that they had different numbers of target domain samples, 4125, 75, 89, and 3235 respectively. Based on the generated quality, Weather 2 and 5 belong to the first category, while Weather 3 and 4 fall into another category. Surprisingly, their performance improvements in the task were roughly similar, with Weather 4, even having only 89 domain samples, achieving a higher improvement in average precision (AP) compared to Weather 2 and 5, which had a greater number of samples. This suggests that for the landmark detection task, the number of domain samples and the generation quality do not solely determine the improvement in AP performance.

As shown in Fig. 7(b), the mixed domains of Weather 2, 3, 4, and 5 exhibit a high degree of overlap with the domain distribution of the test set. Among these four, the mixed domain of Weather 3 differs from the test set's domain distribution in regions 5 and 6 of Fig. 7(b). The approximate alignment of the mixed domain distribution with the test set distribution implies that they are independent and identically distributed. In such cases, the model can better learn the common features of the data and apply them to new unseen samples. Therefore, despite generating images of acceptable quality, accordi, Weather 3 also shows a smaller improvement compared to the other four in the performance of HRNet-w48.

The domain distribution of DayTime 2, as shown in the Fig. 7(b), is completely different from the domain distribution of the test set on the far left. There are also differences between the left side of the overlapping region and the domain distribution of the test set. DayTime 8 exhibits a similar pattern. Looking at the AP results in Table VI and Table VII, it can be observed that the improvements for both DayTime

TABLE VI
AVERAGE PRECISION (AP) AND AVERAGE RECALL (AR) OF SPG-GT WITH HRNET-W32 AS THE BACKBONE ON SLAD.

Methods	Dataset	AP	AP50	AP75	AP_M	AP_L	AR	AR50	AR75	AR_M	AR_L	Time
DayTime 1	test	0.584	0.935	0.640	0.222	0.599	0.668	0.946	0.734	0.329	0.684	0.146
	val	0.593	0.945	0.643	0.248	0.607	0.672	0.952	0.736	0.329	0.688	0.158
DayTime 2	test	0.567	0.922	0.604	0.229	0.580	0.655	0.933	0.715	0.337	0.670	0.143
	val	0.577	0.934	0.637	0.255	0.591	0.661	0.947	0.733	0.324	0.677	0.146
DayTime 3	test	0.574	0.932	0.620	0.183	0.590	0.666	0.941	0.729	0.293	0.684	0.138
	val	0.589	0.934	0.642	0.240	0.605	0.672	0.942	0.738	0.324	0.689	0.134
DayTime 4	test	0.580	0.933	0.636	0.200	0.595	0.667	0.941	0.732	0.308	0.684	0.145
	val	0.597	0.936	0.661	0.234	0.615	0.678	0.949	0.748	0.312	0.695	0.152
DayTime 5	test	0.576	0.922	0.627	0.188	0.592	0.666	0.938	0.734	0.295	0.683	0.143
	val	0.588	0.934	0.638	0.204	0.602	0.674	0.945	0.739	0.302	0.692	0.153
DayTime 6	test	0.579	0.933	0.630	0.189	0.594	0.669	0.945	0.733	0.290	0.687	0.133
	val	0.594	0.934	0.657	0.218	0.611	0.675	0.943	0.744	0.302	0.692	0.151
DayTime 7	test	0.583	0.933	0.643	0.250	0.596	0.672	0.943	0.745	0.353	0.687	0.135
	val	0.595	0.936	0.649	0.252	0.611	0.675	0.948	0.748	0.343	0.692	0.153
DayTime 8	test	0.568	0.923	0.604	0.205	0.582	0.656	0.936	0.715	0.291	0.674	0.144
	val	0.580	0.934	0.626	0.251	0.593	0.665	0.946	0.728	0.361	0.679	0.153
Weather 1	test	0.581	0.934	0.631	0.203	0.595	0.667	0.945	0.732	0.310	0.684	0.139
	val	0.587	0.932	0.644	0.238	0.602	0.671	0.942	0.734	0.312	0.688	0.147
Weather 2	test	0.582	0.934	0.632	0.202	0.597	0.670	0.945	0.737	0.304	0.687	0.141
	val	0.591	0.933	0.656	0.201	0.607	0.670	0.942	0.741	0.278	0.688	0.151
Weather 3	test	0.575	0.924	0.624	0.210	0.591	0.661	0.940	0.724	0.301	0.678	0.141
	val	0.585	0.936	0.639	0.233	0.601	0.667	0.949	0.737	0.324	0.684	0.150
Weather 4	test	0.584	0.934	0.641	0.235	0.600	0.669	0.943	0.741	0.328	0.685	0.140
	val	0.588	0.945	0.644	0.249	0.601	0.669	0.952	0.731	0.324	0.686	0.134
Weather 5	test	0.578	0.933	0.631	0.208	0.594	0.667	0.940	0.734	0.306	0.684	0.136
	val	0.585	0.935	0.632	0.224	0.600	0.668	0.944	0.728	0.308	0.685	0.153
Mix-descent	test	0.588	0.923	0.654	0.195	0.604	0.670	0.936	0.740	0.289	0.688	0.135
	val	0.599	0.935	0.665	0.241	0.614	0.678	0.949	0.750	0.345	0.694	0.152
Mix-rise	test	0.603	0.923	0.663	0.214	0.618	0.689	0.940	0.757	0.304	0.707	0.134
	val	0.618	0.936	0.690	0.233	0.633	0.694	0.944	0.762	0.314	0.712	0.156
Mix-all	test	0.597	0.932	0.646	0.227	0.613	0.688	0.947	0.749	0.322	0.705	0.115
	val	0.616	0.945	0.686	0.242	0.631	0.696	0.950	0.760	0.320	0.714	0.127
Vanilla	test	0.559	0.931	0.589	0.199	0.572	0.646	0.942	0.703	0.305	0.663	0.121
	val	0.574	0.946	0.596	0.207	0.589	0.655	0.951	0.709	0.271	0.673	0.119

2 and DayTime 8 are minimal across two different backbone networks. In fact, DayTime 2 even shows a decrease in AP performance by 0.35% for the HRNet-w48 backbone network. Comparing DayTime 2 and DayTime 8 in Table V, it can be seen that DayTime 2 belongs to the fourth category of data, while DayTime 8 belongs to the third category. Therefore, the generated domain of DayTime 8 is more reliable, which leads to performance improvements for both HRNet-w32 and HRNet-w48.

In addition to that, we observed that in SPG-GT with HRNet-w48 as the backbone, there is a decrease in AP scores when domain transfer data are introduced. Specifically, this decrease is observed for Weather 3 as well as DayTime 2 and 3. In terms of data quality, these domains belong to the fourth and first categories, respectively. This indicates that the decrease in AP scores cannot be attributed to the quality of the generated data alone. Additionally, their distributions align well with the target domain, making their domain transformations more reliable.

To validate the scenario of mixed training among different domains, the 3 categories were combined to form a mixed descent domain, namely Mix-descent. The remaining 10 categories were used to form Mix-rise, and all 13 categories were combined to form Mix-all. Through the 3 multi-domain transfer mixing experiments, as shown in Fig. 7(c), it can be observed that the three mixed domains have similar distributions in regions 2 and 3.

From the two tables, it can be seen that even with the combination of three domains that have minimal or even negative effects on the networks, the SPG-GT achieves an AP increase of 2.7% and 1.15% on two different backbones, respectively, with Mix-rise and Mix-all showing positive ef-

TABLE VII
AVERAGE PRECISION (AP) AND AVERAGE RECALL (AR) OF SPG-GT WITH HRNET-W48 AS THE BACKBONE ON SLAD.

Methods	Dataset	AP	AP50	AP75	AP_M	AP_L	AR	AR50	AR75	AR_M	AR_L	Time
DayTime 1	test	0.600	0.932	0.660	0.208	0.616	0.687	0.941	0.753	0.312	0.704	0.157
	val	0.594	0.945	0.653	0.220	0.610	0.684	0.952	0.752	0.324	0.701	0.165
DayTime 2	test	0.583	0.923	0.631	0.219	0.599	0.672	0.940	0.735	0.328	0.688	0.159
	val	0.589	0.935	0.654	0.220	0.606	0.674	0.944	0.745	0.302	0.691	0.163
DayTime 3	test	0.598	0.933	0.653	0.193	0.613	0.688	0.945	0.752	0.299	0.707	0.151
	val	0.599	0.934	0.650	0.211	0.614	0.689	0.947	0.753	0.318	0.707	0.179
DayTime 4	test	0.595	0.921	0.650	0.204	0.612	0.682	0.940	0.745	0.310	0.700	0.155
	val	0.610	0.936	0.673	0.209	0.624	0.684	0.942	0.749	0.294	0.702	0.164
DayTime 5	test	0.594	0.932	0.660	0.182	0.610	0.683	0.946	0.752	0.282	0.702	0.155
	val	0.601	0.936	0.657	0.211	0.618	0.685	0.947	0.755	0.300	0.704	0.158
DayTime 6	test	0.588	0.929	0.641	0.209	0.604	0.681	0.942	0.747	0.309	0.699	0.159
	val	0.596	0.936	0.659	0.191	0.613	0.682	0.946	0.759	0.290	0.701	0.176
DayTime 7	test	0.598	0.933	0.664	0.228	0.615	0.683	0.942	0.751	0.327	0.701	0.156
	val	0.603	0.938	0.653	0.208	0.620	0.684	0.948	0.743	0.300	0.702	0.173
DayTime 8	test	0.587	0.922	0.637	0.218	0.603	0.673	0.935	0.740	0.312	0.690	0.151
	val	0.596	0.936	0.651	0.250	0.611	0.675	0.946	0.737	0.341	0.691	0.176
Weather 1	test	0.597	0.935	0.663	0.212	0.615	0.684	0.943	0.755	0.302	0.702	0.147
	val	0.603	0.935	0.657	0.234	0.618	0.687	0.947	0.754	0.329	0.704	0.182
Weather 2	test	0.591	0.921	0.643	0.186	0.606	0.681	0.940	0.742	0.290	0.700	0.161
	val	0.604	0.936	0.661	0.231	0.619	0.685	0.949	0.741	0.302	0.703	0.165
Weather 3	test	0.587	0.922	0.631	0.206	0.603	0.677	0.940	0.739	0.304	0.694	0.155
	val	0.603	0.945	0.654	0.209	0.617	0.680	0.950	0.741	0.302	0.698	0.161
Weather 4	test	0.598	0.934	0.660	0.224	0.614	0.683	0.943	0.752	0.326	0.700	0.152
	val	0.612	0.947	0.693	0.214	0.629	0.689	0.951	0.761	0.302	0.708	0.181
Weather 5	test	0.596	0.931	0.652	0.174	0.612	0.682	0.941	0.746	0.277	0.701	0.153
	val	0.598	0.935	0.664	0.241	0.615	0.683	0.942	0.755	0.310	0.701	0.174
Mix-descent	test	0.595	0.931	0.645	0.221	0.609	0.682	0.942	0.741	0.318	0.699	0.156
	val	0.607	0.934	0.671	0.229	0.623	0.688	0.941	0.759	0.302	0.706	0.174
Mix-rise	test	0.601	0.929	0.658	0.205	0.618	0.686	0.943	0.747	0.312	0.704	0.159
	val	0.608	0.923	0.679	0.226	0.623	0.691	0.938	0.763	0.324	0.708	0.167
Mix-all	test	0.605	0.930	0.667	0.225	0.620	0.689	0.946	0.753	0.325	0.707	0.134
	val	0.620	0.937	0.685	0.283	0.634	0.699	0.949	0.767	0.376	0.714	0.136
Vanilla	test	0.585	0.931	0.637	0.191	0.601	0.671	0.942	0.736	0.295	0.689	0.133
	val	0.594	0.936	0.655	0.228	0.609	0.675	0.947	0.748	0.306	0.692	0.146

fects. Furthermore, on the Mix-all mixed domain, the SPG-GT with HRNet-w48 as the backbone achieves the AP score of 62% in validation set, surpassing other algorithms than HRFormer-Base, including the Transformer-based HRFormer-Small, while also demonstrating advantages in inference speed.

Based on the experimental results, we have demonstrated that domain transfer data augmentation significantly improves detection accuracy in landmark detection tasks with limited data. Even in cases where certain domains have very few samples, the use of multiple domains in a complementary manner can still lead to a significant improvement in accuracy.

V. CONCLUSION

This paper proposes SPG-GT, which combines GNNs and Transformers for ship landmark detection. SPG-GT leverages ship structural prior to guide the detection of landmarks. Additionally, we expand the SLAD dataset to create SLAD++ through attribute domain transfer using ship images annotated with timestamps and weather information. Quantitative results on both the existing SLAD dataset and the newly created SLAD++ dataset demonstrate the effectiveness of SPG-GT. Visual results demonstrate that SPG-GT suppresses the interference information and improves the predictions of landmarks, providing explanatory insights into its operational mechanism. It is worth mentioning that the current work focuses only on the RGB images of ship images. In the future, we plan to extend our research to other types of ship data, such as 3D point clouds, infrared image and remote sensing images.

REFERENCES

[1] E. Gundogdu, B. Solmaz, V. Yücesoy, and A. Koc, "Marvel: A large-scale image dataset for maritime vessels," in *Computer Vision–ACCV 2016: 13th Asian Conference on Computer Vision, Taipei, Taiwan,*

November 20-24, 2016, Revised Selected Papers, Part V 13. Springer, 2017, pp. 165–180.

[2] B. Debaque, M. C. Florea, N. Duclos-Hindié, and A.-C. Boury-Brisset, "Evidential reasoning for ship classification: Fusion of deep learning classifiers," in *2019 22th International Conference on Information Fusion (FUSION)*. IEEE, 2019, pp. 1–8.

[3] Z. Shao, L. Wang, Z. Wang, W. Du, and W. Wu, "Saliency-aware convolution neural network for ship detection in surveillance video," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 30, no. 3, pp. 781–794, 2019.

[4] Y. Shan, X. Zhou, S. Liu, Y. Zhang, and K. Huang, "Siamfpn: A deep learning method for accurate and real-time maritime ship tracking," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 31, no. 1, pp. 315–325, 2020.

[5] Q. Zhang, M. Zhang, J. Liu, X. He, R. Song, and W. Zhang, "Unsupervised maritime vessel re-identification with multi-level contrastive learning," *IEEE Transactions on Intelligent Transportation Systems*, 2023.

[6] Q. Bi, M. Wang, Y. Huang, M. Lai, Z. Liu, and X. Bi, "Ship collision avoidance navigation signal recognition via vision sensing and machine forecasting," *IEEE Transactions on Intelligent Transportation Systems*, 2023.

[7] M. Zhang, Q. Zhang, R. Song, P. L. Rosin, and W. Zhang, "Ship landmark: An informative ship image annotation and its applications," *IEEE Transactions on Intelligent Transportation Systems*, 2024.

[8] S.-E. Wei, V. Ramakrishna, T. Kanade, and Y. Sheikh, "Convolutional pose machines," in *Proceedings of the IEEE conference on Computer Vision and Pattern Recognition*, 2016, pp. 4724–4732.

[9] J. Wang, K. Sun, T. Cheng, B. Jiang, C. Deng, Y. Zhao, D. Liu, Y. Mu, M. Tan, X. Wang *et al.*, "Deep high-resolution representation learning for visual recognition," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 43, no. 10, pp. 3349–3364, 2020.

[10] Y. Xu, J. Zhang, Q. Zhang, and D. Tao, "ViTPose: Simple vision transformer baselines for human pose estimation," in *Advances in Neural Information Processing Systems*, 2022.

[11] M. Zhang, Q. Zhang, R. Song, P. L. Rosin, and W. Zhang, "Ship landmark: An informative ship image annotation and its applications," *IEEE Transactions on Intelligent Transportation Systems*, pp. 1–16, 2024.

[12] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, and I. Polosukhin, "Attention is all you need," *Advances in neural information processing systems*, vol. 30, 2017.

[13] J. Devlin, "Bert: Pre-training of deep bidirectional transformers for language understanding," *arXiv preprint arXiv:1810.04805*, 2018.

[14] T. B. Brown, "Language models are few-shot learners," *arXiv preprint ArXiv:2005.14165*, 2020.

[15] N. Carion, F. Massa, G. Synnaeve, N. Usunier, A. Kirillov, and S. Zagoruyko, "End-to-end object detection with transformers," in *European conference on computer vision*. Springer, 2020, pp. 213–229.

[16] Z. Liu, Y. Lin, Y. Cao, H. Hu, Y. Wei, Z. Zhang, S. Lin, and B. Guo, "Swin transformer: Hierarchical vision transformer using shifted windows," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2021, pp. 10 012–10 022.

[17] Z. Li, Y. Zhong, R. Song, T. Li, L. Ma, and W. Zhang, "Detal: Open-vocabulary temporal action localization with decoupled networks," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2024.

[18] Z. Tang, M. Naphade, S. Birchfield, J. Tremblay, W. Hodge, R. Kumar, S. Wang, and X. Yang, "Pamtri: Pose-aware multi-task learning for vehicle re-identification using highly randomized synthetic data," in *Proceedings of the IEEE/CVF international conference on computer vision*, 2019, pp. 211–220.

[19] W. Ding, S. Li, G. Zhang, X. Lei, and H. Qian, "Vehicle pose and shape estimation through multiple monocular vision," in *2018 IEEE International Conference on Robotics and Biomimetics (ROBIO)*. IEEE, 2018, pp. 709–715.

[20] P. Li, H. Zhao, P. Liu, and F. Cao, "Rtm3d: Real-time monocular 3d detection from object keypoints for autonomous driving," in *European Conference on Computer Vision*. Springer, 2020, pp. 644–660.

[21] H. Yu, Y. Xu, J. Zhang, W. Zhao, Z. Guan, and D. Tao, "Ap-10k: A benchmark for animal pose estimation in the wild," *arXiv preprint arXiv:2108.12617*, 2021.

[22] P. C. Bala, B. R. Eisenreich, S. B. M. Yoo, B. Y. Hayden, H. S. Park, and J. Zimmermann, "Automated markerless pose estimation in freely moving macaques with openmonkeystudio," *Nature communications*, vol. 11, no. 1, p. 4560, 2020.

[23] A. Gosztolai, S. Günel, V. Lobato-Ríos, M. Pietro Abrate, D. Morales, H. Rhodin, P. Fua, and P. Ramdya, "Liftpose3d, a deep learning-based approach for transforming two-dimensional to three-dimensional poses in laboratory animals," *Nature methods*, vol. 18, no. 8, pp. 975–981, 2021.

[24] E. Insafutdinov, L. Pishchulin, B. Andres, M. Andriluka, and B. Schiele, "DeeperCut: A deeper, stronger, and faster multi-person pose estimation model," in *ECCV*, 2016, pp. 34–50.

[25] Z. Cao, T. Simon, S.-E. Wei, and Y. Sheikh, "Realtime multi-person 2d pose estimation using part affinity fields," in *CVPR*, 2017, pp. 7291–7299.

[26] E. Insafutdinov, M. Andriluka, L. Pishchulin, S. Tang, E. Levinkov, B. Andres, and B. Schiele, "ArtTrack: Articulated multi-person tracking in the wild," in *CVPR*, 2017, pp. 6457–6465.

[27] U. Iqbal, A. Milan, and J. Gall, "PoseTrack: Joint multi-person pose estimation and tracking," in *CVPR*, 2017, pp. 2011–2020.

[28] A. Newell, Z. Huang, and J. Deng, "Associative embedding: End-to-end learning for joint detection and grouping," *Advances in neural information processing systems*, vol. 30, 2017.

[29] S. Jin, X. Ma, Z. Han, Y. Wu, W. Yang, W. Liu, C. Qian, and W. Ouyang, "Towards multi-person pose tracking: Bottom-up and top-down methods," in *ICCV PoseTrack Workshop*, vol. 2, no. 3, 2017, p. 7.

[30] G. Papandreou, T. Zhu, L.-C. Chen, S. Gidaris, J. Tompson, and K. Murphy, "PersonLab: Person pose estimation and instance segmentation with a bottom-up, part-based, geometric embedding model," in *ECCV*, 2018, pp. 269–286.

[31] Z. Cao, G. Hidalgo, T. Simon, S.-E. Wei, and Y. Sheikh, "Openpose: Realtime multi-person 2d pose estimation using part affinity fields," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 43, no. 1, pp. 172–186, 2021.

[32] S. Jin, W. Liu, W. Ouyang, and C. Qian, "Multi-person articulated tracking with spatial and temporal embeddings," in *CVPR*, 2019, pp. 5664–5673.

[33] H.-S. Fang, S. Xie, Y.-W. Tai, and C. Lu, "Rmpe: Regional multi-person pose estimation," in *Proceedings of the IEEE international conference on computer vision*, 2017, pp. 2334–2343.

[34] K. He, G. Gkioxari, P. Dollár, and R. Girshick, "Mask r-cnn," in *Proceedings of the IEEE international conference on computer vision*, 2017, pp. 2961–2969.

[35] S. Huang, M. Gong, and D. Tao, "A coarse-fine network for keypoint localization," in *ICCV*, 2017, pp. 3028–3037.

[36] G. Papandreou, T. Zhu, N. Kanazawa, A. Toshev, J. Tompson, C. Bregler, and K. Murphy, "Towards accurate multi-person pose estimation in the wild," in *CVPR*, 2017, pp. 4903–4911.

[37] Y. Chen, Z. Wang, Y. Peng, Z. Zhang, G. Yu, and J. Sun, "Cascaded pyramid network for multi-person pose estimation," in *CVPR*, 2018, pp. 7103–7112.

[38] W. Liu, J. Chen, C. Li, C. Qian, X. Chu, and X. Hu, "A cascaded inception of inception network with attention modulated feature fusion for human pose estimation," in *AAAI*, 2018.

[39] B. Xiao, H. Wu, and Y. Wei, "Simple baselines for human pose estimation and tracking," in *ECCV*, 2018, pp. 466–481.

[40] J. Li, C. Wang, H. Zhu, Y. Mao, H.-S. Fang, and C. Lu, "CrowdPose: Efficient crowded scenes pose estimation and a new benchmark," in *CVPR*, 2019, pp. 10 863–10 872.

[41] J. Li, S. Bian, A. Zeng, C. Wang, B. Pang, W. Liu, and C. Lu, "Human pose regression with residual log-likelihood estimation," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2021, pp. 11 025–11 034.

[42] Y. Yuan, R. Fu, L. Huang, W. Lin, C. Zhang, X. Chen, and J. Wang, "Hrformer: High-resolution vision transformer for dense predict," *Advances in Neural Information Processing Systems*, vol. 34, 2021.

[43] R. Liu, J. Lehman, P. Molino, F. Petroski Such, E. Frank, A. Sergeev, and J. Yosinski, "An intriguing failing of convolutional neural networks and the coordconv solution," *Advances in neural information processing systems*, vol. 31, 2018.

[44] J.-Y. Zhu, T. Park, P. Isola, and A. A. Efros, "Unpaired image-to-image translation using cycle-consistent adversarial networks," in *Proceedings of the IEEE international conference on computer vision*, 2017, pp. 2223–2232.

[45] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," *arXiv preprint arXiv:1412.6980*, 2014.

[46] S. Liu, Z. Zeng, T. Ren, F. Li, H. Zhang, J. Yang, C. Li, J. Yang, H. Su, J. Zhu *et al.*, "Grounding dino: Marrying dino with grounded pre-training for open-set object detection," *arXiv preprint arXiv:2303.05499*, 2023.

[47] A. Kirillov, E. Mintun, N. Ravi, H. Mao, C. Rolland, L. Gustafson, T. Xiao, S. Whitehead, A. C. Berg, W.-Y. Lo *et al.*, "Segment anything," *arXiv preprint arXiv:2304.02643*, 2023.

[48] J. C. V. Guerra, Z. Khanam, S. Ehsan, R. Stolkin, and K. McDonald-Maier, "Weather classification: A new multi-class dataset, data augmentation approach and comprehensive evaluations of convolutional neural networks," in *2018 NASA/ESA Conference on Adaptive Hardware and Systems (AHS)*. IEEE, 2018, pp. 305–310.

[49] B. Zhao, X. Li, X. Lu, and Z. Wang, "A cnn–rnn architecture for multi-label weather recognition," *Neurocomputing*, vol. 322, pp. 47–57, 2018.

[50] M. Heusel, H. Ramsauer, T. Unterthiner, B. Nessler, and S. Hochreiter, "Gans trained by a two time-scale update rule converge to a local nash equilibrium," in *Advances in Neural Information Processing Systems*, I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett, Eds., vol. 30. Curran Associates, Inc., 2017, p. 6629–6640.

[51] C. E. Shannon, "The transmission of information," *Bell System Technical Journal*, vol. 27, no. 3, pp. 379–423, 1948.

[52] Z. Wang, A. C. Bovik, H. R. Sheikh, and E. P. Simoncelli, "Image quality assessment: from error visibility to structural similarity," *IEEE transactions on image processing*, vol. 13, no. 4, pp. 600–612, 2004.

[53] R. Zhang, P. Isola, A. A. Efros, E. Shechtman, and O. Wang, "The unreasonable effectiveness of deep features as a perceptual metric," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2018, pp. 586–595.

[54] M. Contributors, "OpenMMLab Pose Estimation Toolbox and Benchmark," https://github.com/open-mmlab/mmpose, 2020.

[55] A. Toshev and C. Szegedy, "Deeppose: Human pose estimation via deep neural networks," in *2014 IEEE Conference on Computer Vision and Pattern Recognition*, 2014, pp. 1653–1660.

[56] A. Radford, J. W. Kim, C. Hallacy, A. Ramesh, G. Goh, S. Agarwal, G. Sastry, A. Askell, P. Mishkin, J. Clark *et al.*, "Learning transferable visual models from natural language supervision," in *International conference on machine learning*. PMLR, 2021, pp. 8748–8763.

[57] R. R. Selvaraju, M. Cogswell, A. Das, R. Vedantam, D. Parikh, and D. Batra, "Grad-cam: Visual explanations from deep networks via gradient-based localization," in *Proceedings of the IEEE international conference on computer vision*, 2017, pp. 618–626.