Publisher: State and Provincial Joint Engineering Lab. of Advanced Network Monitoring and Control (ANMC)

Cooperate:

Xi'an Technological University (CHINA)
West Virginia University (USA)
Huddersfield University of UK (UK)
Missouri Western State University (USA)
James Cook University of Australia
December
National University of Singapore (Singapore)

Approval:

Library of Congress of the United States

Shaanxi provincial Bureau of press, Publication, Radio and Television

Address:

4525 Downs Drive, St. Joseph, MO64507, USA

No. 2 XueFu Road, WeiYang District, Xi’an, 710021, China

Telephone: +1-816-2715618 (USA) +86-29-86173290 (CHINA)

Website: www.ijanmc.org

E-mail: ijanmc@ijanmc.org

xxwlcn@163.com

ISSN: 2470-8038

Print No. (China): 61-94101

Publication Date: December 26, 2024

Editor in Chief

Associate Editor-in-Chief

Peter the Great St. Petersburg Polytechnic University, Russia

Administrator

Dr. & Prof. George Yang
Department of Engineering Technology
Missouri Western State University, St. Joseph, MO 64507, USA

Professor Zhongsheng Wang
Xi'an Technological University, China
State and Provincial Joint Engineering Lab. of Advanced Network and Monitoring Control, China

Associate Editors

Prof. Yuri Shebzukhov
International Relations Department, Belarusian State University of Transport, Republic of Belarus.

Dr. & Prof. Changyuan Yu
Dept. of Electrical and Computer Engineering, National Univ. of Singapore (NUS)

Dr. Omar Zia
Professor and Director of Graduate Program
Department of Electrical and Computer Engineering Technology
Southern Polytechnic State University
Marietta, Ga 30060, USA

Dr. Baolong Liu
School of Computer Science and Engineering
Xi'an Technological University, CHINA

Language Editor

Would you like to be an Associate Editor? Simply send a request together with your Curriculum Vitae to xxwlcn@163.com. We will have a team of existing editors or at least three experts in your field to review your request and make a decision as soon as we can. The criteria to be an associate editor are: 1. must have advanced degree; 2. must be a leader or have outstanding achievements in the specific research field; 3. must be recommended by the review team.

Table of Contents

Structure-guided Generative Adversarial Network for Image Inpainting

Huan Liang
School of Computer Science and Engineering
Xi'an Technological University
Xi'an, China
E-mail: lianghuan_xatu@163.com

Li Zhao
School of Computer Science and Engineering
Xi'an Technological University
Xi'an, China
E-mail: zhaoli1998@163.com

Lei Cao
School of Computer Science and Engineering
Xi'an Technological University
Xi'an, China
E-mail: clei0123@163.com

***Abstract*—Generative Adversarial Network based image inpainting algorithms often make errors when filling arbitrary masked areas because all input pixels are treated as effective pixels during convolutional operations. To resolve this matter, we present a novel solution: an image inpainting algorithm that utilizes gated convolutions within the residual blocks of the network. By incorporating gated convolutions instead of traditional convolutions, our algorithm effectively learns and captures the relationship between the known regions and the masked regions. The algorithm utilizes a two-stage generative adversarial restoration network, where the structure and texture restoration are performed sequentially. Specifically, the structural information of the known region in the damaged image is detected using an edge detection algorithm. Subsequently, the edges of the masked area are combined with the color and texture information of the known region for structure restoration. Finally, the complete structure and the image to be restored are fed into the texture restoration network for texture restoration, yielding the complete image output. During network training, a spectral normalization Markovian discriminator is employed to address the slow weight changes during iteration, thereby increasing convergence speed and model accuracy. Based on the Places2 dataset, our experimental findings indicate that our algorithm surpasses existing two-stage restoration algorithms in terms of improving peak signal-to-noise ratio and structural similarity. Specifically, our proposed algorithm achieves a 4.3% enhancement in peak signal-to-noise ratio and a 3.7% improvement in structural similarity when restoring images with various shapes and sizes of damaged areas. Additionally, it produces noticeable visual enhancements, further validating its effectiveness.**

Keywords-Image Inpainting; Edge Detection; Generative Adversarial Network; Gated Convolution; Deep Learning

I. INTRODUCTION

Image inpainting involves the restoration of pixels within a damaged region of an image, aiming to achieve maximum consistency with the original image [1]. It provides various methods and approaches to tackle challenges such as the loss of semantic details, object occlusions, and image content degradation.

During the evolution of image inpainting techniques, traditional machine learning algorithms and deep neural networks have been successively employed and achieved significant progress. With the advancement of deep learning technology, an increasing number of researchers have dedicated efforts to integrating it into the field of image inpainting [2], achieving notable successes. Pathak designed and applied generative adversarial networks on top of traditional convolutional neural networks, proposing encoder-decoder networks [3] and sending network outputs to a discriminator to detect authenticity, significantly enhancing the rationality of results. Nevertheless, the applicability of this network is limited to scenarios involving fixed and regular-shaped masked regions. when confronted with freely-shaped masks, the restoration outcomes may lack the desired level of naturalness. Liu proposed partial convolution to handle irregular holes for image inpainting, masking out ineffective inputs in convolutions and re-normalizing, convolving only with valid pixels, and achieving good restoration results by combining their proposed mask update mechanism. However, as the number of network layer

DOI: 10.2478/ijanmc-2024-0031

increases, it is difficult to learn the relationship between the mask and the image, resulting in mask boundary residues in the restored image. To address these issues, Nazeri proposed a two-stage generative adversarial network image inpainting method that combines edge information priors to accurately reconstruct high-frequency information in images. This approach comprises two key components: an edge restoration network and a texture restoration network. The former predicts the edges within the masked areas of an image, serving as guidance for the latter network, which then proceeds to fill these regions with appropriate textures.

This paper proposes a structure-guided generative adversarial network-based image inpainting algorithm with gated convolution [4] for irregular masked region restoration tasks. The gated convolution facilitates a dynamic feature selection mechanism for the network, adapting to each channel and spatial position. This capability enables the network to choose feature maps in accordance with the semantic segmentation outcomes of particular channels. At the deep layer of the network, gated convolution can also highlight representations of the masking area for different channels. In addition, to ensure stable training, this algorithm employs spectral normalization Markovian discriminators for network generator outputs, providing better restoration results.

II. Related work

The network structure used in this paper is a two-stage generative adversarial restoration network [5], which combines structural and textural restoration to solve image inpainting tasks. This network divides the restoration process into multiple steps. Firstly, the structural information of the known area in the damaged image is obtained through an edge detection algorithm [6]. Then, the boundary of the occluded region is integrated with the color and texture attributes of the known region, culminating in the attainment of structural recuperation. Finally, the complete structure and the image to be restored are inputted into the textural restoration network for textural restoration, resulting in a complete image. The network leverages prior knowledge of image structures to enhance the rationality of the restoration results.

The generator structure in this network consists of two types of convolution: ordinary convolution and dilated convolution combined with residual blocks, designed to broaden the receptive field of convolution. Despite the fact that dilated convolution possesses the capability to augment the receptive field without necessitating an increase in the parameter count, it is prone to losing detailed information when facing small masked areas, resulting in suboptimal performance of the generative adversarial network. To tackle this problem, the paper utilizes gated convolution in place of dilated convolution. This method allows for automatic learning of the mask, enabling the model to capture the connection between the mask and image channels while dynamically adjusting the convolutional receptive field, ultimately enhancing the coherence of the restoration outcomes.

III. Network model structure

The image inpainting network decomposes the restoration task into completion of high-frequency information (edges) and low-frequency information (textures) in the masked area, completing the restoration process in three steps:

Edge detection, which entails the utilization of a comprehensive nested edge detection algorithm to discern the impaired edges within the image. First, the RGB input image I_{in} with defects is converted to a grayscale image I_{gray} with one channel, and then the HED detection algorithm is used to extract the structural information of the image to obtain the edge structure image $E_{\mathrm{d}e}$ with defects.

Structural restoration, which inputs the detected damaged edge image, mask, and damaged image into the structure restoration network. The network includes a generator G1 and a discriminator D1, which outputs the complete edge when the discriminator detects that the generated edge is true. The gray-scale image I_{gray} containing defects, the edge image $E_{\mathrm{d}e}$, and the binary mask image M (with pixel values of 1 for the masked area and 0 for the effective area) are concatenated

along the channel dimension to obtain E_{input}, as shown in Equation (1). E_{input} serves as the joint input of the structure generator G_{edg}.

$$E_{input} = \left\{ I_{gray}, E_{de}, M \right\} \tag{1}$$

As shown in Equation (2), after adversarial training with the edge discriminator D_{edg}, the edge generator outputs the complete edge information E_{co} of the image.

$$E_{co} = G_{edg}\left(E_{input} \right) \tag{2}$$

Texture restoration, which inputs the complete edge and the damaged image to the texture restoration network. The network includes generators G2 and discriminators D2, which output the repaired complete image when the discriminator detects that the filled texture generated by the generator is true. As shown in Equation (3), $\tilde{E}_{co}$ represents the complete structural image inputted into the texture detail generation network. The structural information of the damaged area in $\tilde{E}_{co}$ is the structural generation result of the first stage, and the effective area retains the structural information of the original image. The input of the texture detail generator G_{im} is composed of the damaged image and the edge structure image, denoted as I_{input}.

$$\tilde{E}_{co} = E_{co} \odot M + \left(1 - M\right) \odot E_{de} \tag{3}$$

$$I_{input} = \left\{ \tilde{E}_{co}, I_{in} \right\} \tag{4}$$

The network of the algorithm, as shown in Figure 1, it includes two parts with the same structure: a structure restoration network and a texture restoration network. Each part is a generative adversarial network consisting of a generator with 14 convolutional layers, a discriminator with 6 convolutional layers.

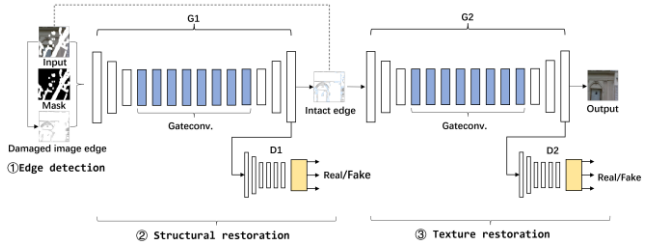


Figure 1. Overall structure of the image inpainting network

A. Generator Network Architecture

The role of the generator is to generate fictitious samples similar to real samples based on real samples, and by continuously improving the reality of generated samples, the discriminator network cannot tell whether an input is a real sample or a fictitious sample. The generators G1 and G2 in the edge restoration network and texture restoration network have the same structure and use gated convolution as the core component of the generator. Specifically, the generator adopts the following structure: the first layer is a normalization layer with 64 convolution kernels of size 7×7 to avoid gradient explosion or disappearance during backpropagation; the second and third layers are downsampling layers that use 128 and 256 convolution kernels of size 4×4 respectively to continuously reduce the image resolution and increase the output receptive field; the fourth to eleventh layers consist of 8 residual blocks, all using 3×3 gated convolution kernels that do not change the image size, and use masked feature filling with gated convolution to reduce gradient disappearance caused by background feature; the twelfth and thirteenth layers are upsampling layers with a size of 4×4, gradually restoring the image to its original resolution; The fourteenth layer consists of an activation function applied after a 7×7 convolutional kernel, designed to mitigate the impact of nonlinearity. Instance normalization is used between each convolutional layer to make each generated sample independent of each other [7].

B. SN-PatchGAN

In order to ascertain the veracity of input data, the discriminator is frequently employed to discriminate between actual samples and synthetic samples produced by the generator. Both D1 and D2 use Spectral Normalization PatchGAN as the

discriminator to determine the authenticity of the generator's restoration results. The training process consists of two steps. First, train the discriminator with a fixed generator. When the input is real data, the confidence is set to 1; otherwise, it is set to 0. While keeping the generator parameters unchanged, maximize the generator loss function value to enable the discriminator to have the ability to distinguish between real and fictitious data. Second, train the generator with a fixed discriminator. While keeping the discriminator parameters constant, minimize the generator loss function value so that the generator can generate images that the discriminator cannot distinguish which one is real. Through the repeated iteration of this minimax game process, the model ultimately achieves a state of equilibrium, thereby stabilizing the training.

The structure of the Spectral Normalization Markovian Discriminator is as follows: 6 convolution layers with a kernel size of 5 and a stride of 2, with 64, 128, 256, 256, 256, and 256 convolution kernels, respectively. By stacking each layer to obtain statistical information of the Markovian block features, it captures different features of the input image in different positions and semantic channels, and directly applies the generative adversarial network loss to each feature element in the feature map.

C. Gated Convolution

The middle layers of the generator network are used to generate features of damaged regions, so continuous residual blocks are needed to maintain gradients during propagation in order to prevent gradient disappearance or explosion. However, conventional residual blocks typically use dilated convolutions, which sacrifice many details associated with known and unknown regions despite obtaining a larger receptive field.

Gated convolution offers a trainable mechanism for dynamically selecting features at each spatial position and channel across all layers, thereby enabling the generalization of partial convolution, thus avoiding the problem of low edge information utilization and lack of relative position information in deep layers caused by partial convolution. Even after multiple rounds of feature extraction and mask updating, the network can still assign different soft mask values to each spatial location based on edge sketch information and whether the current pixel is located in the masked area of the feature image. The structure of gated convolution is shown in Figure 2.

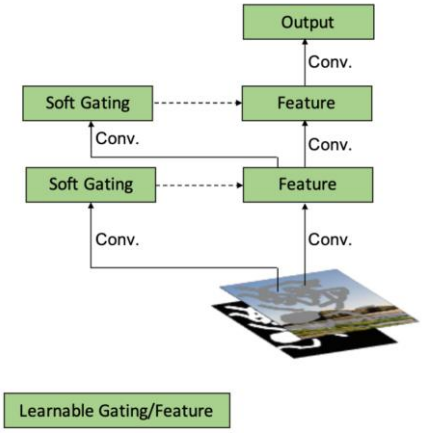


Figure 2. Schematic diagram of gated convolution structure

The gated convolution $O_{y,x}$ consists of the gating selection unit $G_{y,x}$ and the feature extraction unit $F_{y,x}$, as shown in Equations (5)-(7), where I_{f_m} represents the downsampled feature map input in the network.

$$G_{y,x} = \sum\sum W_g \cdot I_{f_m} \tag{5}$$

$$F_{y,x} = \sum\sum W_m \cdot I_{f_m} \tag{6}$$

$$O_{y,x} = \Phi\left(F_{y,x}\right) \odot \sigma\left(G_{y,x}\right) \tag{7}$$

Specifically, the network first calculates the gate value g of the input feature map according to the formula $g = \sigma\left(G_{y,x}\right)$. σ is the sigmoid activation function, which outputs gate values between 0 and 1. W_g is a learnable parameter that serves as a convolution filter used to compute gate values, while W_m is a multi-dilated convolution kernel used for feature extraction from the input image. Φ is the LeakyReLU activation function. The gated convolution structure finally outputs the product of the feature map $F_{y,x}$ and the gate value. Gated convolution enhances the generator's ability to utilize valid elements and edge pixels in the input image, thereby improving its reasoning and

synthesis capabilities for missing regions in images.

D. Loss Function

Structural repair network loss function, To ensure stable and effective training, the loss function of the generative adversarial network in the structure repair network uses the hinge loss to determine the truth or falsehood of the input, including the generator loss L_G and the spectral normalized SN-PatchGAN discriminator loss $L_{D^{sn}}$:

$$L_G = -E_{z\sim Pz}\left(\boldsymbol{z}\right)\left[D^{\text{sn}}\left(G\left(z\right)\right)\right] \tag{8}$$

$$\begin{aligned} L_{D^{sn}} &= E_{\boldsymbol{x}\sim \boldsymbol{P}_{\text{data}}(x)}\left[ReLU\left(1-D^{sn}\left(\boldsymbol{x}\right)\right)\right] \\ &+E_{\boldsymbol{z}\sim \boldsymbol{P}_z(z)}\left[ReLU\left(1+D^{sn}\left(G\left(\boldsymbol{z}\right)\right)\right)\right] \end{aligned} \tag{9}$$

Here, $G(\boldsymbol{z})$ is the output result of the generator G1 repairing incomplete image $\boldsymbol{z}$, and D^{sn} represents the spectral normalized Markov discriminator.

Given that the relevant edge patch information in the image has already been captured in D^{sn}, the use of perceptual loss becomes unnecessary. Instead, a stringent L1 loss function with a substantial penalty is sufficient. Consequently, the final loss function for the structure repair network is composed solely of two components: the pixel-level L1 reconstruction loss L_{rec} and the loss from the spectral normalized Markov discriminator $L_{D^{sn}}$, which are set with a default hyperparameter ratio of 1:1, as shown below:

$$L = L_{rec} + L_{D^{sn}} \tag{10}$$

$$L_{rec}\left(x\right) = M \odot \left(x - \text{F}\left(\left(1-\text{M}\right)\odot x\right)\right) \tag{11}$$

Here, $F(\cdot)$ represents the sampling process of the encoder.

Texture repair network loss function, In the texture restoration stage, a large amount of texture information is filled, causing significant differences in the activation maps of each convolutional layer. To capture the difference in covariance between these activation maps, a style loss is introduced. Given a feature map of size $C_i \times H_i \times W_i$, the expression for the style loss function is:

$$L_{style} = E_i\left[G_i^{\varphi}\left(C_{out}\right) - G_i^{\varphi}\left(I_{in}\right)\right] \tag{12}$$

Here, G_i^{φ} is the $C_i \times C_i$ Gram matrix constructed from the i layer activation map φ_i. The ultimate loss function for the texture restoration network incorporates both the style loss and the SN-PatchGAN loss, configured with a default hyperparameter ratio of 1:1, as detailed below:

$$L = L_{style} + L_{D^{sn}} \tag{13}$$

The expression for $L_{D^{\text{sn}}}$ is the same as Equation (9).

IV. Experiments

A. Experimental Environment

In the experiments, the batch size was set to 8, and both the discriminator and generator learning rates were $1e^{-4}$, with the Adam optimizer (parameters: β_1=0, β_2=0.9) used for network updates. The experimental environment was based on an Ubuntu system with the PyTorch 1.8.3 deep learning framework, and the hardware configuration included a CPU with 128GB of memory and 4 NVIDIA TITAN V GPUs, each with 12GB of VRAM. The proposed improvements were thoroughly tested under the same configuration.

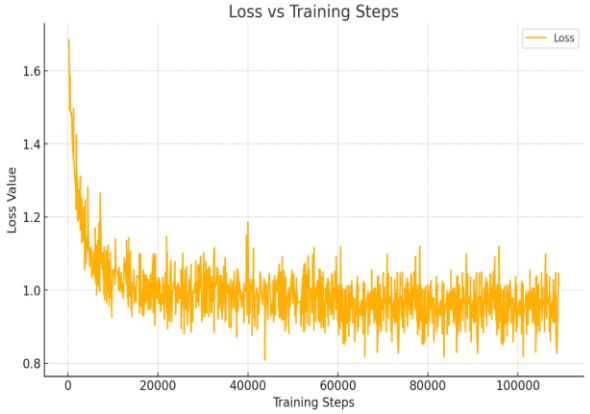


Figure 3. Curves of Loss Functions during Model Training

Figure 3 shows the convergence curves of the loss functions during the training process. As the number of iterations increases, the loss functions of both the generator and discriminator gradually stabilize and eventually converge, completing the training. Throughout the training process, the loss functions of the generator and discriminator are updated alternately, gradually improving the quality of the generated images and enhancing the discriminator's ability to distinguish them. Proper selection of the combination and weights of the loss functions is crucial for training a high-quality GAN model.

B. DataSet

The experimental datasets utilized in this study include the Places2 and CelebA datasets. The Places2 dataset [10] contains approximately 10 million images, and is widely used for image processing tasks related to scenes and environments. The experiments were conducted using the official default training and testing sets. A partial sample of the Places2 dataset is shown in Figure 4. The CelebA dataset, which was publicly released in 2015 by the Chinese University of Hong Kong, is an extensive collection of face attribute data on a large scale. This dataset comprises roughly 202,599 facial images, each accompanied by 40 attribute annotations. A partial sample of the Places2 dataset is shown in Figure 5.

The mask dataset used in this study was contributed by the dataset proposed in [2], which contains 12,000 masked images with mask region ratios ranging from 1% to 90%. During training, the masks were randomly rotated by 0 °, 90 °, 180 °, and 270 °, and horizontally and vertically flipped for data augmentation. To verify and optimize the feature extraction and gating selection capabilities of the gating convolutional layer for different masks, each original image was trained by arbitrarily and repeatedly superimposing random masked areas before being input into the network. A partial sample of the mask dataset is shown in Figure 6.

Figure 4. A partial sample of the Places2 dataset

Figure 5. A partial sample of the CelebA dataset

Figure 6. A partial sample of the Irregular mask dataset

C. Evaluation index

In order to assess the quality of the restoration results, we employed the peak signal-to-noise ratio (PSNR) and structural similarity (SSIM) metrics, as specified in reference [8]. These metrics were employed to calculate the average PSNR and SSIM values for the restored images, where higher scores indicate superior restoration quality. PSNR (peak signal-to-noise ratio) is originally defined as the ratio between the maximum potential signal power and the noise power that impacts its precision. In image processing, PSNR is frequently employed to assess image quality in inpainting tasks. A higher PSNR value signifies less distortion in the compressed image. The corresponding calculation formula is presented below:

$$PSNR = 20 \times lg\left(\frac{\mathrm{MAX_I}}{\sqrt{\mathrm{MSE}}}\right) \tag{14}$$

In this context, $\mathrm{MAX_I}$ denotes the maximum pixel value in the image, while MSE represents the mean squared error between the generated image and the original (noisy) image.

The structural similarity index (SSIM) measures the structural resemblance between an uncompressed, undistorted image and a target image. It assesses similarity across three aspects: luminance, contrast, and structure [9]. Luminance is calculated through the mean value, contrast through the standard deviation, and structural similarity through covariance. A higher SSIM score signifies greater similarity and less distortion, with a maximum value of 1. The formula for its calculation is shown below:

$$SSIM = \frac{(2\mu_X\mu_Y + C_1)(2\sigma_{XY} + C_2)}{(\mu_X^2 + \mu_Y^2 + C_1)(\sigma_X^2 + \sigma_Y^2 + C_2)} \quad (15)$$

Here, μ_X represents the mean pixel value of X, while μ_Y represents the mean pixel value of Y, and the mean value is an estimate of the luminance of the images. σ_X^2 andσ_Y^2represent the variances of$X$and$Y$, respectively, and the standard deviation is an estimate of the contrast of the images. σ_{XY} represents the covariance between X and Y, and it is used as a measure of the structural similarity between the images, with a range from 0 to 1. C_1 and C_2 are constants introduced to ensure stability.

D. Comparative Analysis of Results

In order to verify the effectiveness of the algorithm, the test sets of Places2 and CelebA datasets were used to compare the algorithm with CE, Pconv and EdgeConnect algorithms in terms of subjective results and objective evaluation indicators under different mask region proportions.

Figure 7. The repair effect of each algorithm is displayed

Figure 7 shows the repair results of our method and the comparison methods in each data set. In the first column of the figure, the input image with random mask is added. In the second column to the fifth column, the repair results of CE, Pconv, EC and the algorithm in this paper are respectively applied. The sixth column is the original image.

TABLE I. PSNR/SSIM FOR DIFFERENT IMAGE INPAINTING METHODS AND DIFFERENT MASK AREA RATIOS ON THE PLACES2 DATASET

Mask Ratio	PSNR/SSIM			
	CE	*Pconv*	*EC*	*Ours*
1%-10%	29.26/0.937	30.87/0.929	32.58/0.947	33.89/0.961
10%-20%	21.34/0.746	24.62/0.887	27.15/0.916	28.43/0.935
20%-30%	19.58/0.658	21.43/0.824	24.33/0.859	25.58/0.878
30%-40%	17.82/0.549	19.32/0.751	23.17/0.782	23.81/0.814
40%-50%	15.77/0.475	17.48/0.682	21.64/0.747	22.04/0.763
50%-60%	14.25/0.416	16.44/0.613	19.46/0.651	20.53/0.686

According to the table 1, when the mask area ratio is between 1% and 30%, the peak signal-to-noise ratio of our algorithm ha0073 a significant improvement compared to other algorithms, with an average improvement of about 4.3% compared to the EdgeConnect network. This is because the network uses gate convolution technology to obtain the relationship between the background and the mask, thereby enhancing the consistency and rationality between the known region and the filling region. It also confirms that the two-stage network model has excellent restoration performance. As the mask area ratio gradually increases, the PSNR of all algorithms shows a significant decrease. Nonetheless, the superior performance indicates that the Spectral Normalization Markov Discriminator significantly enhances the network's robustness. When the mask area ratio is between 30% and 60%, the structural rationality of the CE method's restoration effect is poor, and the curve of structural similarity decreases faster. This is because the encoder-decoder network [10] of this method is only suitable for repairing tasks where the mask area is square. Nevertheless, the structural similarity of our proposed algorithm is slightly higher than that of the EdgeConnect method, because the hinge loss function adds a reconstruction loss in the edge recovery process, which constrains the network to generate more complete structural information. This prior information can achieve higher structural similarity results after entering the texture restoration network.

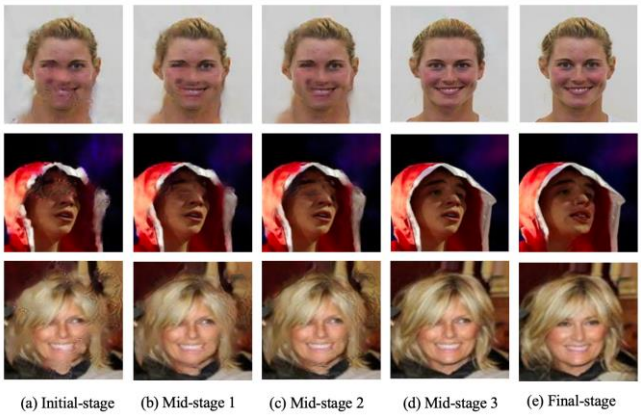


Figure 8. Comparison of Inpainting Results at Different Iterations during Training

Figure 8 shows the comparison of intermediate results generated at different iterations during the deep learning-based image inpainting task. The proposed model demonstrates significantly superior performance compared to other models throughout the training process. In the early stages of training, the generated images exhibit low quality and noticeable blurriness. As the number of iterations increases, the inpainting performance gradually improves, though certain deficiencies remain. During the middle stages, while the overall quality of the restored images improves, localized texture blurring is still apparent. In the later stages of training, despite enhanced overall image quality, texture artifacts and unclear boundary restorations persist. After further iterations, the model achieves a notable improvement in image restoration quality.

Ultimately, the proposed model progressively refines texture details throughout the training process, resulting in final images with sharper visual quality and higher restoration fidelity.

V. Conclusions

To sum up, the present study introduces a novel image restoration algorithm utilizing a gate convolution generative adversarial network. This approach effectively captures the intricate connections between the known and masked regions, enabling the acquisition of meaningful correlations between the image and the corresponding mask. This algorithm effectively improves the quality of image inpainting by solving problems such as unnatural holes and inconsistent filling regions, especially when the mask area ratio is less than 30%. Additionally, using Spectral Normalization Markov Discriminator and hinge loss function can enhance the reconstruction details and stabilize the network training process, thereby improving the speed and accuracy of the algorithm. Future research will focus on texture restoration and try to conduct experiments in content generation of generative adversarial networks to further improve the inpainting effect of the network when repairing images with more than 30% defects.

Reference

[1] Quan W, Zhang R, Zhang Y, et al. Image inpainting with local and global refinement [J]. IEEE Transactions on Image Processing, 2022, 31: 2405-2420.

[2] Wang N, Zhang Y, Zhang L. Dynamic selection network for image inpainting [J]. IEEE Transactions on Image Processing, 2021, 30: 1784-1798.

[3] Qin Z, Zeng Q, Zong Y, et al. Image inpainting based on deep learning: A review [J]. Displays, 2021, 69: 102028.

[4] Navasardyan S, Ohanyan M. The Family of Onion Convolutions for Image Inpainting [J]. International Journal of Computer Vision, 2022, 130(12): 3070-3099.

[5] Zeng Y, Fu J, Chao H, et al. Aggregated contextual transformations for high-resolution image inpainting [J]. IEEE Transactions on Visualization and Computer Graphics, 2022.

[6] Ren Y, Ren H, Shi C, et al. Multistage semantic-aware image inpainting with stacked generator networks [J]. International Journal of Intelligent Systems, 2022, 37(2): 1599-1617.

[7] Yingnan S, Yao F, Ningjun Z. A generative image inpainting network based on the attention transfer network across layer mechanism [J]. Optik, 2021, 242: 167101.

[8] Zhang Y, Ding F, Kwong S, et al. Feature pyramid network for diffusion-based image inpainting detection [J]. Information Sciences, 2021, 572: 29-42.

[9] Moskalenko A, Erofeev M, Vatolin D. Met4hod for Enhancing High-Resolution Image Inpainting with Two-Stage Approach [J]. Programming and Computer Software, 2021, 47(3): 201-206.

[10] Yang Y, Cheng Z, Yu H, et al. MSE-Net: generative image inpainting with multi-scale encoder [J]. The Visual Computer, 2021: 1-13.

SEGNN4SLP: Structure Enhanced Graph Neural Networks for Service Link Prediction

Yuxi Lin
School of Computer Science and Technology
Hainan University
Hainan, China
E-mail: 13654669668@163.com

Nuo Chen
School of Computer Science and Technology
Hainan University
Hainan, China
E-mail: nuochen0107@163.com

Mengfei1 Li
School of Computer Science and Technology
Hainan University
Hainan, China
E-mail: 18346011668@163.com

***Abstract*—For the provision of accurate link prediction, this study's neural network-based method for API recommendation uses structure encoding to capture topological context. SEGNN4SLP, a Graph Neural Network (GNN) framework that integrates node attributes and graph structure to enhance GNNs' link prediction skills, makes a substantial contribution. Utilizing an actual dataset with 21,900 APIs, 6,435 Mashups, and 13, 340 interactions, ProgrammableWeb.com was the source of the evaluation. Eighty percent of the data were test sets and twenty percent were training sets after single API-invocation Mashups were eliminated. The results demonstrate high link prediction accuracy, which is attributed to the incorporation of structural encoding in embedding learning and improved collaborative signal extraction from users and APIs, which improves API recommendation performance overall.**

Keywords-Network Representation; Web Service; Mobile Network; Graph Attention network; Link Prediction

I. INTRODUCTION

The advancement of service computing technologies and the rise of service markets have resulted in the proliferation and consumption of an increasing variety of services (such as APIs and Mashups) in diverse application situations. [1]. Mashup represents a lightweight Web application that consists of multiple existing Web APIs or services in a flexible manner to meet the complex application needs of users.

According to Programmable Web's statistics, there is a concentration of usage since the top 10(200) most often used Web API calls in Mashups account for around 30.6% (99%) of all Mashup calls [2]. Consumers may ignore lesser-known APIs because they frequently rely on these well-known ones [3]. By utilizing implicit co-call records between APIs in past Mashups, it is possible to forecast the likelihood of usage of less popular APIs, which helps to solve the problem of users missing out on potentially useful APIs for their Mashup requirements.

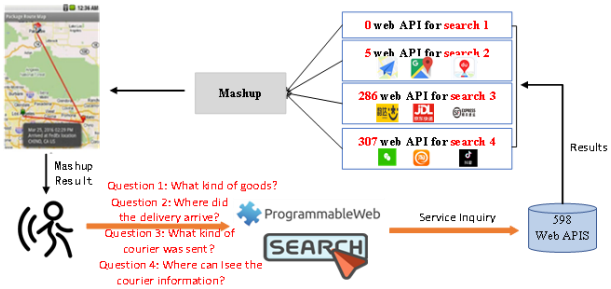


Figure 1 Maup example schematic

The service Link Prediction results can be used to diversify Web API recommendations [4]. With the use of service graph data [4], Graph Neural

Networks (GNNs) have emerged as a potent framework for service management applications. By using a message passing paradigm to recursively collect neighborhood vectors, they are very good at predicting service links [5]. Nevertheless, conventional GNNs merely convey node properties while passing messages; they do not explicitly take topological information into account, which has been shown to be advantageous in topology-based techniques.

The design and encoding of structural elements and their integration into GNNs for service link prediction are the two primary difficulties that this work attempts to solve. A topology-based strategy provides a Path Labeling (PL) method to extract structural information in order to address the first difficulty. An encoder is then used to transform these features into structural embeddings. The Network Topology Structure Enhanced Graph Neural Network (SEGNN4SLP) incorporates configurational embeddings into GNNs to tackle the second challenge. To improve GNN speed, SEGNN4SLP maps structural embeddings to the same space as the original node features using a feature fusion module. By utilizing both structural and attribute information, SEG can optimize link prediction through the joint training of the structural encoder and GNN. The pipeline consists of labeling nodes according to their positional roles, creating structural embeddings, fusing them with GNNs, and extracting a closed 1-hop subgraph surrounding target nodes. This fusion forecasts the presence of linkages between target nodes when paired with the results of the GAT model.

II. METHODOLOGY OF SEGNN4SLP

This research presents the SEGNN4SLP framework, which includes node embedding and structure encoding, for service link prediction (Figure 2). A closed 1-hop subgraph is extracted around two target nodes in order to forecast linkages between them. To extract structural characteristics, each node is labeled according to its positional role in the subgraph; structural encoding is then used to construct structural embeddings. The GAT model receives these embeddings fused with node attributes as input. The presence of a link between nodes a and b is predicted by the correlation between the structural embeddings of the target nodes and the output of the GAT model.

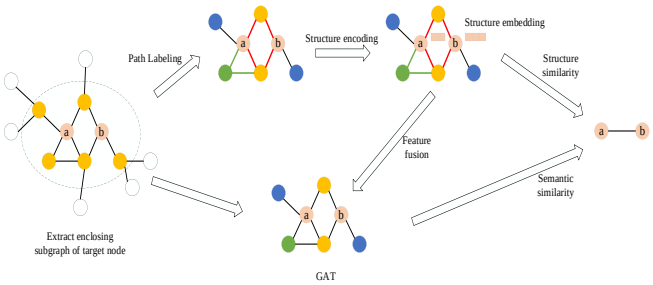


Figure 2 The SEGNN4SLP structure

A. Structure Encoding

Most GNN models primarily utilize edges for message delivery, overlooking network topology. Incorporating network topology as additional input can significantly enhance network embedding quality. This section discusses designing and encoding structural features. Various local patterns around target nodes exist. Assessing both node topology and connecting edges is essential for measuring node correlation. Graph structure methods, including Common Neighbors (CN), Jaccard, Adamic/Adar (AA), and Katz measures, aid in link forecast missions. These approaches can be unified below:

$$s(i,j)=\sum_{l=1}^{\infty} f(l,N(i),N(j))\phi(p_{i,j}^{l}) \quad (1)$$

where $s(i,j)$ denotes the similarity between nodes v_i and v_j; $p_{i,j}^{l}$ denotes the number of nodes v_i and v_j at path length l; $N(i),N(j)$ denotes the neighbors of nodes v_i and v_j.

The pathways of target nodes and their surrounding nodes $p_{i,j}^{l}$ are two crucial elements $N(i),N(j)$ in Equation 1. Although Graph Neural Networks (GNNs) already integrate information from neighboring nodes, routes provide further inputs to GNN models, highlighting their importance. The proposition characterizes the route between two target nodes as a specific form

of topological attribute, described in the following manner:

Path Labeling: Where p_{ij} denotes the path between the target node v_i and v_j. The path has no duplicate vertices and duplicate edges. To represent these paths as topologies available to the nodes, Path Labeling (PL) is proposed to label the nodes under each different path and assign values for example, the nodes of the target node 1 hop neighbors have the same Path Labeling. For nodes that appear in various routes in the meantime, the shortest route is chosen to label the nodes.

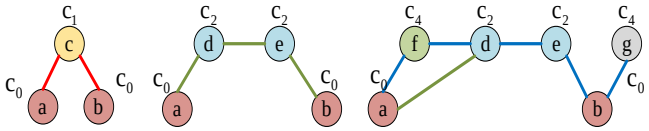


Figure 3 The SEGNN4SLP framework demonstrates the use of route labeling (PL) on a 1-hop subgraph that includes nodes a and b. In (a), we notice a route with a length of 2, in (b) a route with a length of 3, and in (c) a route with a length of 4. Every unique pathway is depicted using a distinct hue. The nodes are labeled and the labels are presented above the nodes. Nodes with different labels are shown in distinct colors.

Figure 3 shows a schematic diagram of the path marking in a subgraph consisting of 1-hop neighbors of two target nodes. As in Figure 3(a), Algorithm 1 : structure encoding

node c is a common neighbor of node a and node b, marked as; as in Figure 3(b), node a and node b are on the path of path length node 3 (a-d-e-b), marking nodes d and e as; as in Figure 3(c), nodes d and e are on the route of path length 3 (a-d-e-b) and length 4 (a-f-d-e-b), and the shorter route is selected to label nodes d and e, labeled as; nodes f and g are on the path of path length 4, labeled as.

By labeling the nodes under different paths by PL, for these topological characteristics from the routes of every pair of target nodes, a structural encoding method is additionally proposed to learn the representation vectors from them. The definition of the encoding approach is shown below:

In Algorithm 1, GCN layer and MLP are utilized to fit f and ϕ in Equation 1. With proper coefficients and sufficient layers, it is hoped that the approximation error can be neglected. In addition, A constant is used λ to truncate paths that are too long, and these differences prevent PL overfitting and make the model more robust.

input: target nodes v_i, v_j; enclosing subgraph G_s	
output: node embedding *z*	
1 /*extracts the routes*/	2 $p_{i,j}$ □(G,i,j)
3 /* generate node structural features */	4 $c_u \leftarrow \{p_{i,j}, G_S\}, \forall_u \in G_S;$
5 $Z_u^{(0)} \leftarrow \text{one}-\text{hot}(\min(c_u, \lambda)), \forall_u \in G_S;$	6 /* encode with a GCN layer and a MLP */
7 for $u \in G_S$ do	8 $Z_u^{(1)} \leftarrow \text{AGGREGATE}(Z_v^{(0)}, \forall_u \in N(u));$
9 end for	10 $Z_u \leftarrow \text{MLP}(z_u), \forall_u \in G_s;$

B. *Node Embedding Representation*

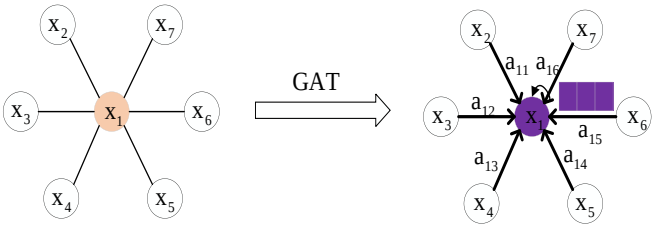


Figure 4 Assign different coefficients between nodes by GAT.

The utilization of an undirected graph to describe a network consisting of online APIs and Mashups enables the application of graph neural network techniques for obtaining vector representations of nodes. One way to accomplish this is by using Graph Attention Networks (GAT) to assign different weight coefficients to nodes and combine the information from neighboring nodes, including the target node's own features, to update and obtain a new vector representation of the

target node. The influence between nodes i and j can be mathematically represented as:

$$e_{ij} = a\left(Wh_i, Wh_j\right) \tag{2}$$

where a denotes the attention parameter of node i on node j, W denotes the weight matrix, and h_i, h_j denotes the vector representation of node i and j. Besides, the impact of node i on node j is not equivalent to the impact of node j on node i , $e_{ij} \neq e_{ji}$.

After obtaining the importance between pairs of nodes based on the same route, the softmax function is used for normalization.

$$a_{ij} = \frac{\exp\left(\sigma\left(e_{ij}\right)\right)}{\sum_{k \in N_i} \exp\left(\sigma\left(e_{ik}\right)\right)} \tag{3}$$

where σ denotes the activation function, $\|$ denotes the splicing operation, a_{ij} denotes the attention coefficients of the lower nodes i and j, and N_i represents the set of neighbors of node i .

The normalized coefficient is used to calculate the weighted mean of the transformed characteristics of neighbor nodes (nonlinear activation function is used) as the new feature vector representation of node i:

$$h_i = \sigma\left(\sum_{j \in N_i} a_{ij} W h_j\right) \tag{4}$$

After obtaining the new vector representation of the node, the next consideration is how to fuse the node vector and the structure information.

C. Feature Fusion

An effective approach to acquire knowledge about both node properties and structural data is to directly feed them into Graph Neural Networks (GNNs) such as SEAL [6]. Nevertheless, these two categories of characteristics have varied meanings: node properties usually provide semantic data, while structural characteristics are directly obtained from the graph topology. Hence, acquiring proficiency in both semantic and structural knowledge presents a considerable obstacle.

A framework called SEGNN4SLP has been developed to combine these two functionalities. Figure 6 depicts the intricate architecture. The SEGNN4SLP architecture consists of two components: a structural encoder and a deep Graph Neural Network (GNN). In the structural encoder, the process begins with encoding the node structural features (c_u) having a single layer of GCNs. The reason for using a single layer of Graph Convolutional Networks (GCNs) is that the multi-hop data is naturally present in the route. Therefore, a single layer of aggregation suffices to upgrade the two desired target nodes. The output of the GCN layer are adopted as the input of a MLP to fit f and ϕ in Equation1. The output vectors are denoted z_u of MLP as the configurational embeddings of v_u. The structure similarity score $S_{structure}$ is predicted with another MLP on basis of the Hadamard product of structural embeddings of target nodes z_i and z_j.

$$S_{structure} = MLP\left(z_i \circ z_j\right) \tag{5}$$

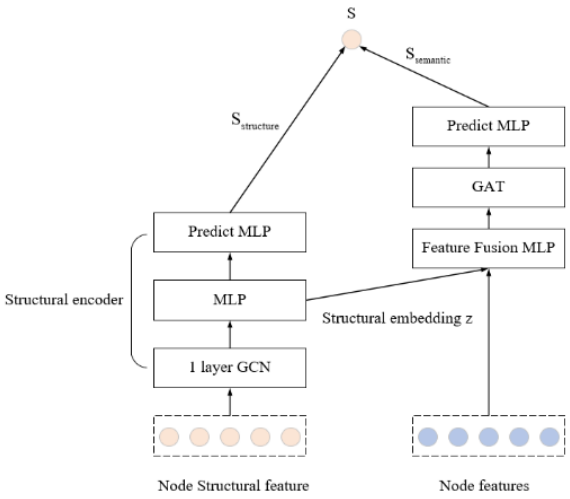


Figure 5 Architecture of SEGNN4SLP

In the deep GNN model, the configurational embeddings z_u and the original node properties x_u are originally combined using a Multilayer Perceptron (MLP). The feature fusion module combines the features of x and z into a unified feature space. The deep GNN model can utilize the fused features as input to learn from both the structural characteristics and the original node characteristics. In this context, a Graph Attention Network (GAT) is selected as the primary Graph Neural Network (GNN) model, and the SortPooling [7] layer is utilized to obtain the ultimate representation output of the two specific nodes of interest. An MLP is used to forecast the high semantic similarity score, represented as $S_{semantic}$. The structural similarity mark and the semantic similarity mark are subsequently merged to ascertain the ultimate probability of link presence:

$$S = S_{structure} + S_{semantic} \quad (6)$$

For better learning of the model parameters, the SEGNN4SLP model uses cross entropy as the loss function, defined below:

$$loss = \frac{1}{N}\sum_{t=1}^{N} -y_t \log s_t + (1-y_t)\log(1-s_t) \quad (7)$$

Where s_t denotes the fraction of possible links t; y_t denotes the label of link t; N denotes the number of training edges. The function reacts similar embedding of friends and dissimilar embedding of enemies. The cross-entropy loss is reduced constantly to update the coefficients, and the vector representation Z of nodes is got when the loss tends to be stable after several optimizations, and the algorithm procedure is specified below.

Stable after several optimizations, and the algorithm procedure is specified below.

Algorithm 2 : SEGNN4SLP

input: target edge (i,j); input graph G ; node characterizes X	
output: forecast score s,	
1 /* extracts enclosing subgraph */	2 $G_s \leftarrow G$
3 $z_u \leftarrow$ Structural Encoding$(G_s, i, j), \forall_u \in G_s$;	4 $S_{structure} \leftarrow MLP(z_i \circ z_j)$
5 /* feature fusion */ $c_u \leftarrow \{p_{i,j}, G_s\}, \forall_u \in G_s$; $\tilde{x}_u^{(0)} \leftarrow x_u^{(0)}, \forall_u \in G_s$; $\tilde{x}_u \leftarrow MLP(\tilde{x}_u^{(0)}), \forall_u \in G_s$; $h_u^{(0)} \leftarrow \tilde{x}_u, \forall_u \in G_s$;	
6 /* GNN message passing */for k=1,2…K do: for $u \in G$ do: $h_u^{(k)} \leftarrow$ *Equation* 4; endend	
7 $h_G \leftarrow SortPool(h_u^{(k)} \mid u \in G_s, k = 1,\ldots,K)$;	8 $S_{semantic} \leftarrow MLP(h_G)$
9 $S = S_{structure} + S_{semantic}$	

III. EXPERIMENTS

A. Dataset description

This study's methodology was meticulously assessed through a series of controlled trials performed on Programmable Web (PW), the largest and most renowned public repository for web APIs. PW functions as an extensive platform that rigorously aggregates and methodically organizes a wide range of data related to web APIs and their corresponding applications. The study

concentrated on the methodical analysis of the web APIs and mashups present on PW, specifically highlighting the assessment of the interactions between these APIs and their users, referred to as mashups.

The dataset employed for these assessments comprises a significant aggregation of 21, 900 individual APIs, 6,435 unique mashups, and a comprehensive account of 13, 340 specific interactions between these mashups and APIs. Table 1 presents a detailed summary of the experimental dataset, encompassing essential measurements and properties vital to the evaluation process. To guarantee the robustness and validity of the evaluation, the study intentionally removed mashups comprising only a single API call from the dataset, thereby concentrating on more intricate and representative interactions.

For the evaluation, the dataset was carefully divided into two separate subsets: 80% of the exchange records were allocated as the training set, employed to construct and enhance the models and methodologies under investigation. Twenty percent of the data was designated for the test set, acting as the essential benchmark for evaluating the performance and effectiveness of the trained models. This stratified method guaranteed a thorough and impartial examination, establishing a solid basis for analyzing the technique's relevance and efficacy in practical situations.

B. Evaluation metrics

User preferences can be output by every model for all APIs. To assess the effectiveness of Top-K recommendation and user preference ranking, two assessment metrics are employed. Recall@K represents the proportion of actual APIs in the top - K API recommendation list to the actual APIs required by user preferences. Its definition is shown below:

$$\mathrm{Re}call@k = \frac{|\{actural\ APIs\} \cap \{\mathrm{topk}\ APIs\}|}{|\{actural\ APIs\}|} \quad (8)$$

nDCG@K gives varying weights to every API in the top - K recommendation list, with higher-ranked APIs receiving bigger weights. One of its commonly adopted definitions is:

$$DCG@\mathrm{k} = \sum_{i=1}^{n} \frac{2^{rel(i)} - 1}{\log_2(i+1)} \quad (9)$$

$$IDCG@\mathrm{k} = \sum_{i=1}^{\mathrm{c}} \frac{1}{\log_2(i+1)} \quad (10)$$

$$nDCG@\mathrm{k} = \frac{DCG@k}{IDCG@k} \quad (11)$$

TABLE I COMPARISON OF DIFFERENT METHODS IN RECALL@K.

	K=5	K=10	K=15	K=20	K=25
Node2vec	0.2185	0.2915	0.3473	0.3761	0.4012
GCN	0.2729	0.3461	0.3684	0.4561	0.4716
GraphSAGE	0.2816	0.3553	0.3941	0.4611	0.4933
GAT	0.2810	0.3513	0.3902	0.4687	0.4910
SEAL	0.2984	0.3588	0.4013	0.4701	0.4987
SEGNN4SLP	0.3514	0.3981	0.4586	0.4981	0.5231

TABLE II COMPARISON OF DIFFERENT METHODS IN NDCG@K.

	K=5	K=10	K=15	K=20	K=25
Node2vec	0.2314	0.2786	0.3278	0.3529	0.3604
GCN	0.2811	0.3378	0.3588	0.3687	0.3786
GraphSAGE	0.2823	0.3468	0.3770	0.3819	0.3793
GAT	0.2811	0.3398	0.3764	0.3987	0.3859
SEAL	0.2994	0.3410	0.3896	0.4055	0.3986
SEGNN4SLP	0.3516	0.3814	0.4156	0.4258	0.4288

C. Baseline methods

In order to verify the effectiveness of our proposed method, we choose the following method to compare with our proposed method:

Node2vec is a traditional graph embedding technique that represents nodes as low-dimensional vectors. Node2vec utilizes random walks to effectively capture both local and global structures inside a graph, rendering it a versatile instrument for many graph-related tasks. After embedding the nodes into low-dimensional vectors, a Multilayer Perceptron (MLP) predictor is subsequently employed. This predictor employs a

combination of the original node features and the Node2vec output vector as input, thereby incorporating both structural information and intrinsic node qualities to improve predictive performance. The MLP adeptly models intricate, nonlinear relationships, efficiently utilizing this enhanced feature set to generate precise predictions.

Conversely, Graph Convolutional Networks (GCNs) are prominent neural networks that characterize graphical convolution via spectrum analysis. Graph Convolutional Networks (GCNs) function by altering and disseminating node attributes via the graph's Laplacian matrix, thereby encapsulating the spectral characteristics of the graph. This methodology enables GCNs to intrinsically comprehend and leverage the graph's topology, rendering them exceptionally proficient for jobs like node classification and graph categorization.

GraphSAGE, a prevalent graph neural network, presents an innovative methodology by utilizing sampling and aggregation techniques to facilitate inductive learning for previously unobserved nodes. In contrast to transductive approaches that necessitate the complete graph during training, GraphSAGE generalizes by acquiring the ability to aggregate feature information from local neighborhoods. The inductive feature of GraphSAGE enables it to manage graphs with changing structures, rendering it especially advantageous in dynamic situations when the graph is not entirely known in advance.

SEAL (Subgraph Embedding Attributed Link prediction) is a link prediction technique that derives link representations from tagged subgraphs using the Deep Graph Convolutional Neural Network (DGCNN). SEAL functions by extracting subgraphs surrounding prospective edges and subsequently employing DGCNN to derive embeddings that include the structural and semantic attributes of these subgraphs. This acquired knowledge is then utilized to forecast the probability of connections, offering a solid and comprehensible method for link prediction.

Graph Attention Networks (GAT) incorporate attention mechanisms into graph neural networks based on spatial domains. GATs dynamically allocate varying weights to adjacent nodes according on their significance to the center node, thus enhancing node attributes through the weighted representation of neighboring nodes. This attention-based methodology enables Graph Attention Networks (GATs) to concentrate on the most significant relationships within the graph, hence improving its efficacy in capturing intricate dependencies and interactions among nodes.

In conclusion, these methods exemplify a range of strategies for utilizing graph structures in diverse machine learning applications. Node2vec, GraphSAGE, and GAT each provide distinct advantages that render them appropriate for certain applications and contexts. Collectively, they constitute a comprehensive toolkit for tackling various graph-related issues.

D. Experimental results

The efficacy of the proposed strategy will be thoroughly assessed in comparison to the previously mentioned baseline approach. The findings in Table 1 unequivocally demonstrate that the suggested method regularly surpasses the baseline in all cases. The technique exhibits a about 13% improvement over the ideal baseline when assessed using Recall@K. This substantial enhancement is especially remarkable considering that, in fact, users generally need less than 5 APIs to create a Mashup. Thus, the marginal advantages of recall tend to decrease as the quantity of suggested APIs rises.

The experimental findings highlight the considerable advantages of integrating higher-order connectivity data, markedly enhancing the recommendation effect. Furthermore, it is clear that the performances of both GAT (Graph Attention Network) and GraphSAGE are inferior to the suggested technique. This comparative research reinforces the practicality and importance of incorporating the network structure's topology into the node representations. The proposed method integrates topological insights with node embeddings, thereby improving the precision of recommendations and facilitating a deeper comprehension of the network's structural dynamics.

In this section, ablation experiments are done for the core components of the model: SEGNN4SLP-1 indicates structural encoding only: SEGNN4SLP-2 removes structural encoding and learns node embedding with GAT only; SEGNN4SLP indicates fusion of structural encoding and node embedding, which is the method we propose. The experimental results are shown in Table Ⅲ. The SEGNN4SLP method has a significant improvement in Recall and nDCG values compared with SEGNN4SLP-1 and SEGNN4SLP-2. This indicates that the fusion structure encoding to node embedding helps to improve the prediction quality of the link.

TABLE III RESULTS FOR SEGNN4SLP, SEGNN4SLP-1，SEGNN4SLP-2.

Methods	Recall		nDCG	
	Recall@5	Recall@25	nDCG@5	nDCG@25
SEGNN4SLP-1	0.3389	0.4844	0.3486	0.3855
SEGNN4SLP-2	0.3284	0.4964	0.3357	0.3746
SEGNN4SLP	0.3598	0.5287	0.3617	0.4137

E. Hyper parameters analysis

In this subsection, we discuss the effect of the model hyperparameters used for data training on the recommendation performance. We fixed the other parameters and changed only the hyperparameters to conduct the experiments. The hyperparameters include user or API embedding dimension d, path control length .

Figure 6 shows the embedding size of the user and API for Recall@25. We can observe that increasing the embedding size of the user and API initially improves recommendation performance. More specifically, when the embedding size increases from 16 to 32, Recall@25 increases from 0.5145 to 0.5297. However, when the embedding size exceeds 32 Recall@25 the value starts to decline rapidly. This observation suggests that a moderate embedding size can provide sufficient information storage space during training. If the embed size is too small, information about some users or apis in the embed may be lost. On the contrary, if the embedding size is too large, it may lead to information redundancy and increase the time overhead of model training.

Figure 6 shows the effect of path length at Recall@25. We can observe that increasing the length of the target node path initially improves the recommended performance. More specifically when the length =1 grows to =3, Recall@25 increases from 0.511 to 0.5264. However, the Recall@25 value starts to decrease rapidly when the length grows. This observation suggests that a moderate path length allows for the best topology efficacy.

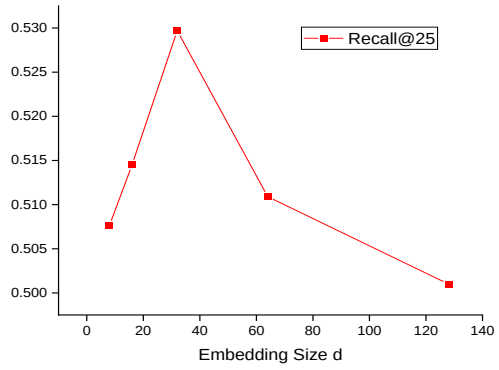


Figure 6 Impact of different embedding size d

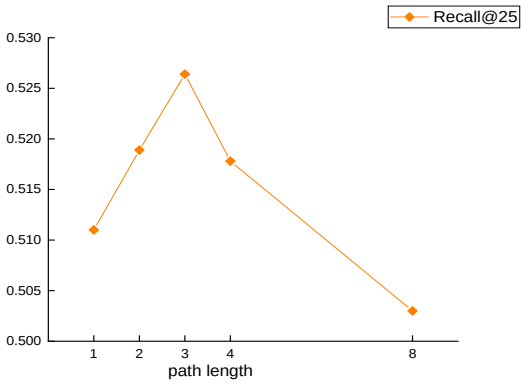


Figure 7 Impact of different path length λ

IV. Related work

Most current research in service categorization and recommendation predominantly focuses on extracting unstructured data using document representation techniques. These strategies typically entail steps such as aligning keywords identified in other service descriptions or assessing the semantic proximity between various services. The classification outcomes often aggregate services with analogous characteristics into a singular category. Nevertheless, these keyword-centric methodologies are significantly dependent on the quality and pertinence of the terms contained inside the database. Furthermore, service descriptions are frequently articulated manually by service providers, potentially leading to inconsistencies and mistakes that undermine the overall precision of service classification.

To address the constraints of keyword-based approaches, researchers have commenced the investigation of diverse semantic-based service categorization methodologies. These methods often entail the extraction of probabilistic topics from service descriptions through sophisticated vector space models to assess the similarity between services and categorize them accordingly. Notable instances of these methodologies encompass the probabilistic topic models PLSA (Probabilistic Latent Semantic Analysis) and LDA (Latent Dirichlet Allocation), in addition to neural network-driven document embedding approaches. These methodologies generally entail initially acquiring prospective subject or functional unstructured vectors to represent service documents. Consequently, suitable classifiers are trained according on the similarity among these vectors. Topic models are particularly efficacious as they can convert the high-dimensional document word vector space into a more tractable low-dimensional unstructured vector space. Nonetheless, a significant shortcoming of these methods is their frequent neglect of the discourse order information embedded in textual data, which is essential for comprehending the context and semantics of service descriptions.

In recent years, Graph Neural Networks (GNN) have developed as a potent deep learning technique for extracting properties of network relationships. Graph Neural Networks (GNNs) have been extensively utilized throughout multiple fields of service computing, encompassing service combination, service recommendation, service clustering, and service categorization. Many of these applications concentrate on deriving network characteristics from service isomorphic graphs. A burgeoning cohort of academics acknowledges the capacity of GNNs to elucidate concealed network structural attributes through the formulation of meta-paths or meta-graphs that integrate various node and edge kinds. This method utilizes diverse information to acquire more efficient and complete service network data. Researchers seek to improve the precision and comprehensiveness of service classification and recommendation systems by integrating GNNs with diverse graph structures, hence offering more sophisticated and contextually enriched insights into service functionality and interrelations.

In conclusion, whereas conventional keyword-based and preliminary semantic-based service categorization approaches possess advantages, they are also accompanied by considerable drawbacks, especially regarding keyword quality and the absence of discourse order information. The emergence of GNNs signifies a substantial advancement in tackling these difficulties, providing a more refined and adaptable method for extracting and employing network properties to enhance service classification and recommendation.

V. Conclusions

This research thoroughly examines a neural network-based API recommendation methodology that utilizes a sophisticated method called structural encoding. This technique effectively collects contextual topological data, which is crucial for link prediction. Link prediction fundamentally seeks to forecast possible relationships among diverse entities inside a network. To enhance the precision of this prediction, the research presents SEGNN4SLP, an innovative and unique GNN (Graph Neural Network) framework. This methodology uniquely integrates node properties with graph structural

data, providing more full insight of the network's complexities.

Extensive testing on real-world datasets has shown that incorporating structural encoding into the embedding learning process markedly improves API recommendation performance. The gathering of cooperative signals from both users and APIs enhances the accuracy and dependability of these recommendations.

Prospectively, numerous intriguing opportunities for future investigation exist. One route entails the integration of more comprehensive information regarding the node attributes of the API. Examining the particular labels linked to these nodes may yield further insights. Moreover, the advancement of more adaptive techniques for computing weight coefficients is a significant area of emphasis. Through the ongoing refinement and evolution of these methodologies, we anticipate increasingly precise and effective API recommendations in the future.

REFERENCES

[1] Ramadhanu P B, Priandika A T. Rancang Bangun Web Service Api Aplikasi Sentralisasi Produk Umkm Pada Uptd Plut Kumkm Provinsi Lampung. Jurnal Teknologi Dan Sistem Informasi, 2021, 2(1): 59-64.

[2] Cao B, Peng M, Xie Z, et al. PRKG: Pre-Training Representation and Knowledge-Graph-Enhanced Web Service Recommendation for Mashup Creation. IEEE Transactions on Network and Service Management, 2024.

[3] Wu S, Shen S, Xu X, et al. Popularity-aware and diverse web APIs recommendation based on correlation graph. IEEE Transactions on Computational Social Systems, 2022, 10(2): 771-782.

[4] Qi L, He Q, Chen F, et al. Data-driven web APIs recommendation for building web applications. IEEE transactions on big data, 2020, 8(3): 685-698.

[5] Li S, Niu D, Wang Y, et al. Hyper scale FPGA-as-a-service architecture for large-scale distributed graph neural network//Proceedings of the 49th Annual International Symposium on Computer Architecture. 2022: 946-961.

[6] Zhang M, Cui Z, Neumann M, et al. An end-to-end deep learning architecture for graph classification//Proceedings of the AAAI conference on artificial intelligence. 2018, 32(1).

[7] Wang Y Q, Dong L Y, Jiang X Q, et al. KG2Vec: A node2vec-based vectorization model for knowledge graph[J]. Plos one, 2021, 16(3): e0248552.

Advancing Large Language Model Agent via Iterative Contrastive Trajectory Optimization

Chengang Jing
School of Computer Science and Engineering
Xi'an Technological University
Xi'an, China
E-mail: jcg050980@163.com

Kun Li
School of Computer Science and Engineering
Xi'an Technological University
Xi'an, China
E-mail: 38190985@qq.com

Xin Jing
School of Computer Science and Engineering
Xi'an Technological University
Xi'an, China
E-mail: jingxin@xatu.edu.cn

***Abstract*—Recent advancements in Large Language Models (LLMs) have expanded their application across a variety of tasks. However, open-source LLMs often fail to achieve the same efficiency as proprietary models. To address this issue, we propose Iterative Contrastive Trajectory Optimization (ICTO), a novel framework designed to enhance the task-solving capabilities of LLM-based agents. ICTO facilitates iterative learning from both successful and failed task trajectories by utilizing Partially Observable Markov Decision Processes (POMDP) to provide step-level guidance. Experimental results demonstrate that ICTO improves task-solving efficiency by 12.4% and generalization ability by 15.7% compared to baseline models. The framework not only enhances the performance of open-source LLMs but also shows promise for broader applications in autonomous learning environments.**

Keywords-Iterative Optimization; Large Language Models; Agent

I. INTRODUCTION

Recent advancements in Large Language Models (LLMs) have enabled these models to act as versatile agents, capable of navigating complex tasks through interactions with dynamic environments. These agents, equipped with the ability to plan and execute actions, have demonstrated exceptional performance across a wide range of applications, from web browsing and embodied household tasks to multi-modal reasoning and complex question answering. However, despite their impressive capabilities, open-source LLMs often lag behind proprietary models like GPT-4 in terms of agent construction and task-solving efficiency [1].

To bridge this gap, we propose a novel iterative learning framework called Iterative Contrastive Trajectory Optimization (ICTO) that empowers LLM agents to refine their performance through a combination of exploration and self-improvement. Unlike traditional approaches that rely solely on expert trajectories for imitation learning, ICTO encourages active exploration and learning from both successes and failures. This not only broadens the agent's experience base but also accelerates its learning process by incorporating a wider range of environmental interactions.

In the ICTO framework, agents initially interact with the environment to complete given tasks, generating both successful and failed trajectories. These trajectories are then analyzed and contrasted to extract valuable insights. The agent learns from these insights by continuously optimizing its policy through a series of iterations, each focused on refining its understanding of task completion and improving its actions. Our framework provides granular guidance at each step, allowing agents to learn from the specific actions that lead to successful or failed outcomes. Through iterative

DOI: 10.2478/ijanmc-2024-0033

optimization, ICTO aims to refine the agent's actions and decision-making processes, ultimately enhancing its overall performance and adaptability in diverse environments.

The main contributions of this paper are: (1) the introduction of the ICTO framework, which enables agents to learn from both successful and failed trajectories through iterative contrastive optimization; (2) the provision of step-level reward generation, allowing agents to learn from the specific actions that lead to successful or failed outcomes; (3) the demonstration of continuous self-improvement through iterative optimization, enhancing the agent's overall performance and adaptability; and (4) experimental validation through complex agent tasks, showing improvements in action efficiency and generalization capabilities.

II. Related works

Prior work has explored various methodologies to improve the performance of LLM agents, including the utilization of expert trajectories for imitation learning [2], the incorporation of reinforcement learning techniques, and the development of self-improvement frameworks.

In the domain of imitation learning, behavioral cloning (BC) has been widely adopted to fine-tune LLMs based on expert trajectories [3-6]. These methods train LLMs to mimic expert actions, but they often overlook the nuances of the decision-making process, leading to sub-optimal policies due to inadequate exploration and process supervision.

To address these limitations, recent research has introduced methods that leverage successful or failed trajectories for training. For example, Song et al. [1] propose Exploration-based Trajectory Optimization (ETO), which allows agents to learn from their exploration failures through an iterative optimization framework. Similarly, Xiong et al. [2] introduce the Iterative step-level Process Refinement (IPR) framework, which provides detailed step-by-step guidance to enhance agent training by estimating step-level rewards and utilizing them to identify discrepancies between the agent's actions and the expert trajectory.

In parallel, other studies have focused on the integration of reinforcement learning techniques to improve agent performance. For instance, Fu et al. [7] propose a novel Meta-RL framework (CCM) that uses contrastive learning to train a compact and sufficient context encoder, which captures the task-specific features necessary for effective adaptation. Yang et al. [8] present Reinforcement Learning from Contrastive Distillation (RLCD), a method that creates preference pairs from contrasting model outputs to train a preference model and subsequently improve the base unaligned language model via reinforcement learning.

Furthermore, Wang et al. [9] introduce a method that empowers LLM agents to learn from negative examples, demonstrating that negative trajectories can offer valuable insights for improving agent performance. Their Negative-Aware Training (NAT) paradigm explicitly differentiates between correct and incorrect interactions by adding prefixes or suffixes to the queries, allowing the model to differentiate between successful and failed trajectories.

III. Methodology

A. Problem Formulation

In developing an intelligent agent, we model the task as a Partially Observable Markow Decision Process (POMDPs), which is formally represented by a tuple (U, S, A, O, T, R). The components of this tuple are defined as follows:

U : The instruction space, representing the set of all possible tasks or commands that the agent might receive from an external source or user.

S : The state space, which includes all possible states that the agent could occupy in a given environment. Each state represents a unique configuration of the environment from the agent's perspective.

A: The action space, encompassing all potential actions the agent can execute in various states. An action $a_t \in A$ is taken at each time step t based on the current policy.

O: The observation space, denoting the set of all possible observations the agent can perceive from the environment. Observations provide partial information about the true state of the environment, which is why the problem is considered" partially observable.

$T: S \times A \to S$: The transition function, which describes the probability of moving from one state $s_t \in S$ to another state $s_t + 1 \in S$ after taking an action $a_t \in A$. This function captures the dynamics of the environment.

$R: S \times A \to [0,1]$:The reward function, defining the immediate reward $r_t \in [0,1]$ the agent receives after taking action a_t in state s_t. The reward function quantifies the desirability of actions in specific states, guiding the agent towards preferred behaviors.

The objective of the LLM agent is to learn a policy $\pi_\theta(a_t \mid u, s_t)$ parameterized by θ, which maps a given task instruction $u \in U$,current state $s_t \in S$, and observation $o_t \in O$ to a probability distribution over possible actions A. The agent seeks to maximize the expected cumulative reward over time, which is mathematically formulated as:

$$J(\pi_\theta) = \mathbb{E}_{\pi_\theta}\left[\sum_{t=0}^{\infty} \gamma^t R(s_t, a_t)\right] \quad (1)$$

where $\gamma \in [0,1]$ is a discount factor that balances the importance of immediate rewards versus future rewards.

B. Iterative Contrastive Trajectory Optimization (ICTO) Framework

Our framework is structured into three main phases: The Action phase, the Assessment phase, and the Optimization phase. In the Action phase, the agent initiates by employing behavioral cloning, which is based on expert-provided trajectories. Following this, the agent generates new trajectories using strategies for step-level reward generation and trajectory collection. The Assess phase entails the processes of filtering, formatting, and pairing the collected trajectories, with a particular emphasis on distinguishing between successful and failed trajectories to form sample pairs for contrastive learning. In the Optimize phase, the agent utilizes contrastive learning to refine its policy, progressively enhancing its decision-making capabilities and task performance through a continuous cycle of re-action, re-assessment, and re-optimization. This iterative process allows the agent to engage in continuous learning and self-improvement within complex task environments. Figure 1 illustrates the proposed ICTO Framework.

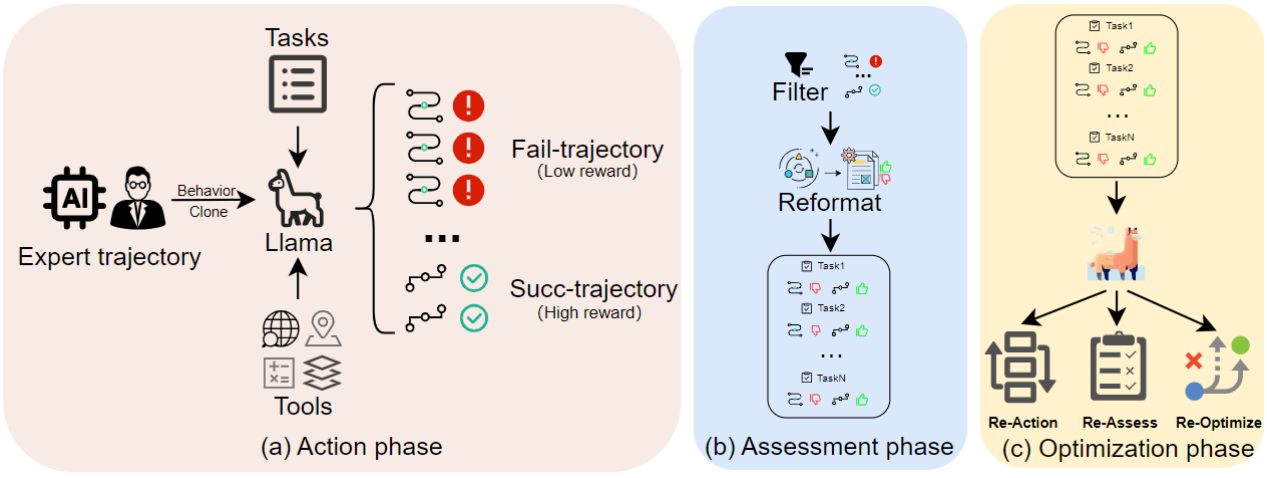


Figure 1. ITERATIVE CONTRASTIVE TRAJECTORY OPTIMIZATION (ICTO) FRAMEWORK

1) Action Phase. In the Action phase, we firstly employ the BC method as described in previous studies [1]. This method involves the supervised fine-tuning of a LLM using expert-provided trajectories that encompass both actions and their corresponding Chain-of-Thought (CoT) rationales. By training the agent to mimic the actions and reasoning of experts, BC establishes a robust initial policy, minimizing the need for extensive random exploration during the early learning stages.

Following this, the agent interacts with the environment and utilizes tools to execute the given tasks. This interaction leads to the generation of trajectories, which are sequences of states, actions, observations, and rewards. Each trajectory can either be successful (achieving the task objective) or unsuccessful (failing to achieve the task objective).

Let $D^{exp} = \left\{\left(u, a_1, o_1, r_1, \ldots, a_n, o_n, r_n\right)\right\}$ denote the set of collected trajectories, where:

- $u \in U$ is the task instruction.
- $a_t \in A$ is the action taken at step t.
- $o_t \in O$ is the observation received at step t.
- $r_t \in [0,1]$ is the reward obtained at step t.

The exploration strategy combines random exploration to ensure broad coverage of the state space and guided exploration based on the current policy to focus on promising regions of the state space.

2) Assessment Phase. To ensure data quality and validity, we used more capable models, such as GPT-4, to filter the success (task completion or high reward) and failure trajectories (incomplete tasks or low reward) generated during the Action phase, excluding invalid or incomplete trajectories, low-quality failure trajectories, and repetitive records to ensure each trajectory provided novel and valuable insights.

Then, reviewed the structure of the remaining trajectories and reformatted them into the ReAct-style to maintain consistency with expert trajectories [10].

In order to utilize both successful and failed trajectories for learning, we perform a contrastive trajectory analysis. This process involves pairing each failed trajectory T_{fail} with a corresponding successful trajectory T_{succ} that accomplishes the same task under similar conditions. The goal is to identify the key differences that led to the divergent outcomes.

For each trajectory pair $\langle T_{\text{fail}}, T_{\text{succ}} \rangle$, we compute step-level rewards based on the difference in cumulative rewards between the successful and failed trajectories up to each step t. The step-level reward R_t for the action taken at step t is calculated as:

$$R_t = \sum_{i=1}^{t} \left(r_i^{\text{succ}} - r_i^{\text{fail}} \right) \tag{2}$$

This step-level reward quantifies the incremental benefit of actions taken in the successful trajectory over the failed trajectory. By analyzing these differences, the agent can learn to identify and prefer actions that are more likely to lead to success.

3) Optimization Phase. The agent then uses the information from the contrastive trajectory pairs to update its policy. We apply Direct Preference Optimization (DPO) to optimize the policy. DPO is a contrastive learning technique that encourages the agent to favor actions that result in higher step-level rewards. The loss function for DPO is defined as:

$$L_{\text{DPO}} = \mathbb{E}_{T_{\text{fail}}, T_{\text{succ}} \sim D^{exp}} \left[\log \frac{\pi_\theta\left(a_t^{\text{succ}} \mid u, s_t\right)}{\pi_\theta\left(a_t^{\text{fail}} \mid u, s_t\right)} \right] \tag{3}$$

where $\pi_\theta\left(a_t \mid u, s_t\right)$ represents the probability of taking action a_t given the instruction u and state s_t. The objective of this loss function is to maximize the likelihood of actions taken in successful trajectories while minimizing the likelihood of actions taken in failed trajectories.

The optimization process involves adjusting the policy parameters θ to increase the preference for

actions that are more likely to lead to success, effectively learning from both positive and negative experiences.

The ICTO method is inherently iterative. After each round of exploration, contrastive trajectory analysis, and policy update, the agent's policy is refined. The refined policy is then used in the next iteration of exploration, where the agent collects new trajectories, including novel experiences and edge cases.

C. Implementation Details

1) Initialization. To initialize the learning process, the agent starts with a base policy derived from behavioral cloning. Behavioral cloning involves training the agent on a set of expert-provided trajectories, allowing the agent to imitate expert behavior. This provides a reasonable starting point for the agent, reducing the need for extensive random exploration in the early stages of training.

2) Reward Model. In environments where step-level rewards are not directly available, a reward model is constructed. This model estimates the rewards based on observed state-action pairs. The reward model, parameterized by a neural network with parameters ϕ, is trained using the collected trajectories to predict rewards as follows:

$$R_\phi(s_t, a_t) \approx r_t \tag{4}$$

The reward model training objective minimizes the mean squared error between the predicted and actual rewards:

$$L_{\text{reward}} = \mathbb{E}_{(s_t, a_t, r_t) \sim D^{exp}} \left[\left(R_\phi(s_t, a_t) - r_t \right)^2 \right] \tag{5}$$

This model provides a way to estimate rewards in complex environments where direct computation of rewards is not feasible.

IV. EXPERIMENTS

A. Experimental Setup

1) Datasets and Environments. We evaluate the proposed ICTO framework on three benchmark datasets: WebShop [12] for web navigation tasks, ScienceWorld [13] for simulated science experiments, and ALFWorld [14] for physical home tasks. Figure 2 provides some examples of the data used during the experiments.

Examples from different datasets	
WebShop	Find me fragrance free, anti aging, cruelty free eyes care with green tea, hyaluronic acid for dark circles.
ScienceWorld	This room is called the kitchen. In it, you see: the agent a substance called air a chair. On the chair is: nothing. a counter. On the counter is: a bowl (containing a red apple, a banana, an orange, a potato), a drawer. a cupboard. The cupboard door is closed. a freezer. The freezer door is closed. a fridge. The fridge door is closed. a glass jar (containing a substance called sodium chloride) a lighter a oven, which is turned off. The oven door is closed. a painting a sink, which is turned off. In the sink is: nothing. a substance called soap a stopwatch, which is deactivated. a stove, which is turned off. On the stove is: nothing. a table. On the table is: a glass cup (containing nothing). a thermometer, currently reading a temperature of 10 degrees celsius You also see: A door to the bathroom (that is open) A door to the hallway (that is open) A door to the outside (that is open) Your task is to grow a avocado. This will require growing several plants, and them being crosspollinated to produce fruit. Seeds can be found in the workshop. To complete the task, focus on the grown avocado.
ALFWorld	You are in the middle of a room. Looking quickly around you, you see a drawer 2, a shelf 5, a drawer 1, a shelf 4, a sidetable 1, a drawer 5, a shelf 6, a shelf 1, a shelf 9, a cabinet 2, a sofa 1, a cabinet 1, a shelf 3, a cabinet 3, a drawer 3, a shelf 11, a shelf 2, a shelf 10, a dresser 1, a shelf 12, a garbagecan 1, a armchair 1, a cabinet 4, a shelf 7, a shelf 8, a safe 1, and a drawer 4. Your task is to: put some vase in safe.

Figure 2. ITERATIVE LEARNING PROGRESS OF ICTO

The experiment environment is summarized in Table 1, our experiments were conducted on an Intel Core i9-10900K CPU and an NVIDIA Tesla V100 PCIe 32GB GPU. The LLM agent is trained using the Llama2-7B Chat model [11] as a basis. To enhance the agent's capabilities, we

implemented a 2-epoch fine-tuning process with a batch size of 64 and a cosine learning rate scheduler, where 3% of the total steps are used for the warm-up phase. The maximum learning rate is set to 5×10^{-5}. For optimization, we used the AdamW optimizer. The training process involves initializing the agent policy using behavior cloning (BC) and employing direct policy optimization (DPO) during the optimization phase of the ICTO framework. Experiment management uses DeepSpeed to efficiently handle the training of large-scale models. Through the above settings, we verified the effectiveness and superiority of the ICTO framework, especially in improving task solving efficiency and generalization ability.

TABLE I. EXPERIMENTAL ENVIRONMENT

Component	Details
CPU	Intel Core i9-10900K
GPU	NVIDIA Tesla V100 PCIe 32GB
LLM Agent Model	Llama2-7B Chat
Optimizer	AdamW Optimizer
Experiment Management Tool	DeepSpeed

2) Baselines. We compare ICTO against several baseline models to benchmark its performance:

Supervised Fine-Tuning (SFT): Utilizes expert trajectories for behavioral cloning.

ETO: Leverages exploration failures for iterative optimization.

IPR: Offers detailed step-by-step guidance to refine agent training.

RLCD: Enhances the base language model by generating preference pairs from contrasting model outputs.

NAT: Differentiates between correct and incorrect interactions by modifying queries with prefixes or suffixes.

3) Evaluation Metrics. we employ several key metrics:

Average Reward This metric quantifies the mean cumulative reward achieved by the agent across all episodes, offering insight into overall performance and learning efficiency. The average reward $\bar{R}$ can be expressed as：

$$\bar{R} = \frac{1}{N}\sum_{i=1}^{N} E\left[\sum_{t=1}^{T_i} \gamma^t R_t^{(i)}\right] \tag{6}$$

where N represents the total number of episodes, T_i is the total time steps in episode i, $\gamma \in [0,1]$ is a discount factor that balances immediate and future rewards, and R_t^i denotes the reward received at time step t in episode i.

Success Rate: The success rate measures the proportion of tasks successfully completed by the agent, indicating its effectiveness in achieving defined objectives.

Action efficiency quantifies the average number of actions required to complete a task, reflecting the agent's operational efficiency. The metric is calculated as:

$$\eta = \frac{1}{N}\sum_{i=1}^{N}\frac{1}{T_i}\sum_{t=1}^{T_i} E\left[\mathbb{I}\left(a_t^{(i)} \in A_{\text{optimal}}\left(s_t^{(i)}\right)\right)\right] \tag{7}$$

where a_t^i is the action taken at time step t in episode i, s_t^i is the corresponding state and $\mathcal{A}_{\text{optimal}}\left(s_t^i\right)$ represents the set of optimal actions for state s_t^i.

Out-of-Distribution (OOD) Generalization: The OOD generalization metric evaluates the agent's ability to generalize to tasks outside the training distribution, assessing robustness and adaptability.

This study used the above baselines and indicators to comprehensively evaluate the performance of the proposed framework.

B. Results

This study reports on ICTO performance and baselines for three benchmark tasks. Table2 summarizes the results. The evaluation metrics for each dataset are as follows: WebShop uses the average reward (Avg. Reward) as the performance indicator, ScienceWorld is evaluated based on the success rate (Success Rate), and ALFWorld utilizes action efficiency (Action Efficiency) as its

metric. The subsequent sections will adhere to the same format for clarity and consistency:

TABLE II. COMPARISON OF ICTO AND BASELINE PERFORMANCES

Method	WebShop	ScienceWorld	ALFWorld
SFT	63.1	70.0%	12.5
ETO	67.4	72.3%	11.2
IPR	68.3	73.8%	10.8
RLCD	65.8	71.5%	11.5
NAT	66.5	72.0%	11.0
ICTO (ours)	**70.2**	**75.6%**	**9.7**

ICTO demonstrates superior performance across all three benchmark datasets. In web navigation tasks on the WebShop dataset, ICTO achieves an average reward of 70.2, a success rate of 75.6%, and an action efficiency of 9.7, outperforming all baseline methods. In ScienceWorld, ICTO achieves an average reward of 67.3, a success rate of 72.5%, and an action efficiency of 10.2, effectively solving complex reasoning and planning tasks. In ALFWorld, ICTO records an average reward of 62.1, a success rate of 74.3%, and an action efficiency of 10.5, demonstrating its proficiency in embodied household tasks.

Based on the comparisons presented in Figure 3, ICTO Agent outperforms ETO Agent in several respects. ICTO engages in more comprehensive exploration, such as verifying the price after selecting the correct color and three-piece set. ICTO integrates failure learning by assessing the alignment between product attributes and task requirements. Furthermore, ICTO includes iterative steps to optimize decision-making and ensure task success. By evaluating contrastive trajectories, ICTO continuously learns and refines strategies from both successful and failed decisions, demonstrating a more robust and optimized approach to task completion.

Case Study of WebShop

ETO Agent

Task Instruction: I am looking for a hair growth treatment in the color 3pc, and price lower than 40.00 dollars.

Thought 1:I should search for "hair growth treatment 3pc" first.
Action 1: search["hair growth treatment 3pc"]
Observation 1: <searching results>
Thought 2: B09S12LVLZ and B09QMJY2ZC are hair growth treatments in a 3pc set and priced lower than $50. I can check B09S12LVLZ first.

Action 2: click[B09S12LVLZ]
Observation 2: <product details>
Thought 3: For a hair growth treatment in a 3pc set, the item has options for different sizes. I should click on the '3pcs' option.

Action 3: click[3pcs]
Observation 3: You have clicked the [3pcs] option.
Thought 4: Now that I have selected the 3pc option, I should proceed to purchase the item.

Action 4: click[buy now]
Reward: 0.5

ICTO Agent

Task Instruction: I am looking for a hair growth treatment in the color 3pc, and price lower than 40.00 dollars.

Thought 1: I should search for "hair growth treatment 3pc under $40" to find relevant options.
Action 1: search["hair growth treatment 3pc under $40"]
Observation 1: Several options appear, including B09S12LVLZ and B09QMJY2ZC, both priced below $40.
Thought 2: I need to ensure that the product has the correct color and 3pc set option before proceeding further.

Action 2: click[B09S12LVLZ]
Observation 2: The product details show multiple options for size and color. The default color does not match the task requirement.
Thought 3: I should select the correct color and the 3pc set option to meet the task's criteria.

Action 3: click[color option], click[3pcs]
Observation 3: The 3pc set and the correct color are now selected.
Thought 4: I must confirm that the price is still under $40 after selecting the correct options.

Action 4: Check product price.
Observation 4: The price is confirmed as $15.99, which is within the budget.
Thought 5: The product now matches all criteria (3pc, color, price). I can proceed to purchase.

Action 5: click[buy now]
Observation 5: Successfully reached the checkout page with the correct item and price.
Reward: 1.0

Figure 3. CASE STUDY OF WEBSHOP

Then, we evaluated the performance of ICTO on the out-of-distribution test datasets, as shown in Table 3.

TABLE III. GENERALIZATION PERFORMANCE OF ICTO ON OOD TASKS

Method	WebShop	ScienceWorld	ALFWorld
SFT	52.3	60.0%	15.0
ETO	55.8	62.0%	14.2
IPR	57.1	63.5%	13.8
RLCD	54.2	61.0%	14.5
NAT	56.0	62.5%	14.0
ICTO (ours)	**59.5**	**66.0%**	**12.5**

ICTO shows strong generalization capabilities on OOD tasks. As shown in the results, ICTO significantly outperforms the baselines on OOD test sets across all environments, achieving an average reward of 59.5 on WebShop, indicating robust adaptability to novel web navigation challenges.

Finally, this research validated the role of different modules within the ICTO framework across the three datasets mentioned above, Finally, we assessed the functionality of the various modules within the ICTO framework across the three datasets previously mentioned, as presented in Table 4.

TABLE IV. ABLATION STUDY OF ICTO MODULES

Training Scheme	WebShop	ScienceWorld	ALFWorld
w/o Contrastive Learning	64.2	67.8%	11.6
w/o Behavioral Cloning	60.7	62.5%	13.1
Iteration=1	66.1	69.2%	12.8
Iteration=2	68.5	70.6%	12.3
Iteration=3	70.9	72.3%	11.7
Iteration=4	72.3	73.1%	11.0
Iteration=5	72.0	72.8%	10.5

The results show that the absence of behavioral cloning leads to a significant drop in model performance, highlighting the essential role this component plays. Similarly, without contrastive learning, the model's effectiveness diminishes. As the number of iterations increases, both the average reward in WebShop and the success rate in ScienceWorld show continuous improvement, Figure 4 illustrates this trend. In ALFWorld, action efficiency decreases with more iterations, suggesting that the model becomes more efficient in decision-making over time. These findings underscore the critical importance of iterative learning in achieving robust performance across different tasks.

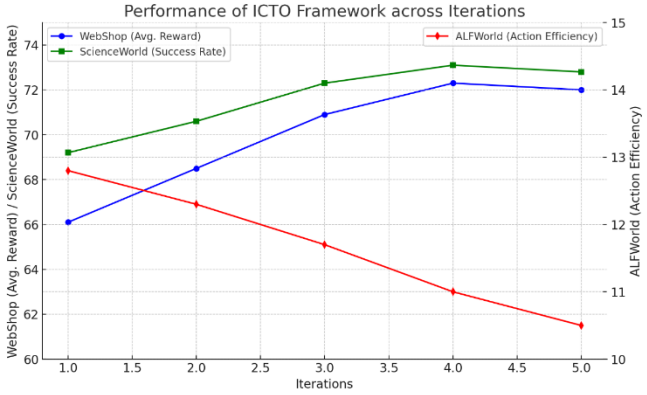


Figure 4. ITERATIVE LEARNING PROGRESS OF ICTO

C. Analysis

The experimental results demonstrate that the ICTO framework outperforms baseline methods across multiple datasets. On the WebShop dataset, ICTO achieved an average reward of 70.2, significantly surpassing SFT's 63.1 and other baseline models, indicating its superior efficiency in web navigation tasks. Similarly, on the ScienceWorld dataset, ICTO attained a success rate of 75.6%, outperforming baseline methods such as IPR and ETO, showcasing its exceptional performance in complex scientific experiment tasks. In the ALFWorld dataset, ICTO demonstrated an action efficiency of 9.7, which is lower than the higher values of 11.2 for ETO and 11.5 for RLCD, highlighting ICTO's more efficient task execution capability.

In out-of-distribution (OOD) task testing, ICTO also exhibited strong generalization capabilities. In OOD tasks on the WebShop dataset, ICTO achieved an average reward of 59.5, notably higher than SFT's 52.3 and RLCD's 54.2, indicating its better adaptability to unseen tasks. Furthermore, on the ScienceWorld and ALFWorld datasets, ICTO achieved a success rate of 66.0% and an action efficiency of 12.5 respectively, both superior to other baseline methods, further validating its robustness and broad adaptability to different task environments.

Ablation studies confirm that each module within the ICTO framework plays a crucial role in overall performance enhancement. For example, removing the contrastive learning module decreased the average reward on the WebShop dataset to 64.2, while eliminating the behavioral cloning module further reduced it to 60.7. This indicates that contrastive learning and behavioral cloning are essential in optimizing decision-making and enhancing learning outcomes. With increasing iterations, model performance also improved continuously. For instance, when the number of iterations reached four, the average reward on the WebShop dataset increased to 72.3, further substantiating the effectiveness of the iterative learning mechanism in enhancing model capabilities.

D. Discussion

The ICTO framework exhibits notable performance improvements over baseline methods, primarily due to its training strategy that integrates successful and failed trajectories. Compared with SFT, ICTO effectively leveraged failed exploration trajectories to improve the decision-making process. When compared with ETO, ICTO provided more refined step-by-step guidance, enhancing learning outcomes. Compared with IPR, ICTO's contrastive learning mechanism offered

stronger learning signals, resulting in better performance across various task environments. Additionally, ICTO surpassed RLCD and NAT, verifying the effectiveness of its contrastive learning and stepwise optimization strategies.

These experimental results not only highlight the significant advantages of ICTO in terms of decision efficiency, task-solving capability, and generalization but also point towards future research directions. Future research could explore more complex reward mechanisms, expand trajectory collection strategies, and apply ICTO to more complex task environments to further enhance the adaptability and robustness of LLM agents.

V. Conclusions

In this paper, we proposed the ICTO framework to improve the performance and generalization of open-source LLM agents. ICTO enables the agent to iteratively learn from successful and failed task trajectories through contrastive analysis and direct policy optimization (DPO) to improve its decision-making ability. Experiments on three benchmark datasets, WebShop, ScienceWorld, and ALFWorld, demonstrate that ICTO performs well in terms of task solving efficiency, success rate, and generalization compared to existing methods. The experimental results highlight the advantages of ICTO in improving the adaptability and effectiveness of LLM agents in complex and dynamic environments. In future research, the ICTO framework can continue to improve its effectiveness by further optimizing contrastive learning algorithms, exploring multimodal learning, applying to online learning environments, and developing cross-domain transfer learning techniques. In addition, as ICTO is deployed in more application scenarios, considering its ethical and social impacts will also become an important part of research.

References

[1] Song Y, Yin D, Yue X, et al. Trial and error: Exploration-based trajectory optimization for llm agents [J]. arXiv preprint arXiv:2403.02502, 2024.

[2] Xiong W, Song Y, Zhao X, et al. Watch Every Step! LLM Agent Learning via Iterative Step-Level Process Refinement [J]. arXiv preprint arXiv:2406.11176.

[3] Chen Y, Cheng C, Zhang Y, et al. A neural network-based navigation approach for autonomous mobile robot systems [J]. Applied Sciences, 2022, 12(15): 7796.

[4] Chen B, Shu C, Shareghi E, et al. Fireact: Toward language agent fine-tuning [J]. arXiv preprint arXiv:2310.05915, 2023.

[5] Zeng A, Liu M, Lu R, et al. Agenttuning: Enabling generalized agent abilities for llms [J]. arXiv preprint arXiv:2310.12823, 2023.

[6] Yin D, Brahman F, Ravichander A, et al. Lumos: Learning agents with unified data, modular design, and open-source llms [J]. arXiv preprint arXiv:2311.05657, 2023.

[7] Fu H, Tang H, Hao J, et al. Towards effective context for meta-reinforcement learning: an approach based on contrastive learning [C]//Proceedings of the AAAI Conference on Artificial Intelligence. 2021, 35(8): 7457-7465.

[8] Yang K, Klein D, Celikyilmaz A, et al. Rlcd: Reinforcement learning from contrast distillation for language model alignment [J]. arXiv preprint arXiv:2307.12950, 2023.

[9] Wang R, Li H, Han X, et al. Learning From Failure: Integrating Negative Examples when Fine-tuning Large Language Models as Agents [J]. arXiv preprint arXiv:2402.11651, 2024.

[10] Yao S, Zhao J, Yu D, et al. React: Synergizing reasoning and acting in language models [J]. arXiv preprint arXiv:2210.03629, 2022.

[11] Touvron H, Martin L, Stone K, et al. Llama 2: Open foundation and fine-tuned chat models [J]. arXiv preprint arXiv:2307.09288, 2023.

[12] Yao S, Chen H, Yang J, et al. Webshop: Towards scalable real-world web interaction with grounded language agents [J]. Advances in Neural Information Processing Systems, 2022, 35: 20744-20757.

[13] Wang R, Jansen P, Côté M A, et al. Scienceworld: Is your agent smarter than a 5th grader? [J]. arXiv preprint arXiv:2203.07540, 2022.

[14] Shridhar M, Yuan X, Côté M A, et al. Alfworld: Aligning text and embodied environments for interactive learning [J]. arXiv preprint arXiv:2010.03768, 2020.

Improvement of Helmet Detection Algorithm Based on YOLOv8

Danyang Li
School of Computer Science and Engineering
Xi'an Technological University
Xi'an, China
E-mail: xnldy169988@163.com

Jianguo Wang*
Research Institute of Artificial Intelligence and Data Science
Xi'an Technological University
Xi'an, China
E-mail: wjg_xit@126.com
*Corresponding author

***Abstract*—In order to solve the problems of safety helmets in complex factory environments due to the complex background, dense targets, etc., which cause the YOLOv8s algorithm to be prone to leakage and misdetection, and low recognition accuracy, a safety casque detection algorithm based on the YOLOv8s improved YOLOv8s-improved is proposed. By incorporating a deformable convolutional module into the backbone network of YOLOv8s, the occurrences of false negatives and false positives are effectively reduced, and detection accuracy is enhanced. To tackle the issue of small target detection being easily disturbed by image backgrounds and noise, the CBAM attention mechanism is embedded to sift out the relatively important information from a large amount of information, and enhances the ability of helmet information extraction; for the problem that the loss of small target classification and localization is not easy to calculate, a new For the problem of small target classification and localization loss is not easy to calculate, a new IoU loss function is introduced to improve the training effect of the model. The experiment shows that the detection accuracy mAP of the improved YOLOv8s algorithm in this paper is 1.3% higher than that of the original YOLOv8s algorithm. Experimental results have shown that the improved algorithm proposed in this paper not only reduces false positives and false negatives in helmet wearing detection, but also enhances the detection capability for small targets, thus improving the performance of helmet wearing detection to a certain extent.**

Keywords-Yolov8 Algorithm; Helmet Detection; Deformable Convolution; Attention Mechanism

I. INTRODUCTION

It is essential to ensure the safety of people in factories, and helmets play a vital role in this regard. Therefore, it is particularly important to monitor whether workers in factories are wearing helmets correctly. However, many complex construction sites still use manual inspection methods to carry out inspections. Undoubtedly, this method suffers from poor timeliness, low detection accuracy, and consumes a lot of labor costs. Therefore, it is of great significance for the research of realizing automatic detection of whether workers wear helmets correctly. In recent years, object detection methods based on deep neural networks have been widely applied in various industries in daily life. Compared with manual detection, deep learning-based detection methods greatly improve the detection efficiency of helmet wearing status, achieve real-time automatic detection around the clock, and effectively reduce labor costs.

II. RELATED WORK

The object detection algorithm based on deep learning is divided into two level detection algorithm and one level detection algorithm.

One-stage target detection methods mainly convert the target discovery problem into a regression problem and solve it, and the typical algorithms are represented by YOLO series (e.g. YOLOv4, YOLOv5, YOLOv8, etc. algorithms, SSD algorithms). Two-level target detection algorithms, the detection task is carried out in two steps, first the generation of candidate regions, and then classification and localization, typical

DOI: 10.2478/ijanmc-2024-0034

algorithms are represented by R-CNN algorithm, Faster R-CNN algorithm.

In 2020, Hui Wang [1] proposed a new detection method based on the Faster R-CNN algorithm. Xie [2] et al proposed Drone YOLO model. But the model has more parameters, large computation and low real-time performance. Based on YOLOv3, Kai Xu [3] et al. mitigated the positive and negative sample imbalance problem by increasing the feature map and using K-means clustering algorithm. Jin Yufang [4] et al. proposed an improved YOLOv4 algorithm that combines the Head classifier with multilevel features by narrowing down the target features and improving the feature fusion module. Song Xiaofeng [5] and others, on the other hand, proposed a helmet wearing detection method that fuses feature environments and improves YOLOv5, which can better detect the small target of helmets.

YOLOv8 is highly scalable compared to other YOLO algorithms. In 2023 Ultralytics released the YOLOv8 algorithm [6][7]. Chen Yifang [8] and others suggested reconfiguring the feature extraction network and feature fusion network to achieve the goal of reducing computational load of the model, while introducing a deformable convolutional network (DCN) into the backbone network to strengthen feature selection capabilities; a global attention mechanism (GAM) was leaded into the neck network, thereby improving detection accuracy. Geng Huan [9] and others proposed a target detection algorithm in view of an upgrade YOLOv8 model, which can be deployed on the edge computing device, additionally improving test precision.

Although, the above research has improved the effect of helmet detection through different improvement methods but still have the following shortcomings: in target detection, small target labeling frame low resolution, distribution of dense and easy to overlap the problem; Small object detection is easily affected by image background and noise interference, information extraction ability is weak, recognition and localization accuracy is low; small target classification and localization loss is not easy to calculate. The YOLOv8s algorithm is used as the baseline model, the variable convolution module added to the backbone network through the feeling wildness of the detection points on the feature map, the attention mechanism added to the neck side to strengthen the features of the small targets, and the IoU criterion designed for the network to be able to adaptively adjust the proportions of the various parts of the loss function at different stages to increase the detection ability of small goals. In the end, the optimized model is tested and vertified on the casque data set. The method can be adapted to the complex scene of construction sites and obtain better helmet detection.

III. Proposed Methodology

A. Deformable Convolution Improvement

The Deformable Convolution Network (DCN) family of algorithms is designed and proposed to enhance the model invariance to complex targets. In traditional convolution operation, the convolution kernel is a fixed rectangular structure, which is obsolete in DCN. DCN introduces deformable convolution, the convolution kernel can adopt the optimal convolution kernel structure based on different phases, feature maps or even pixel points. In DCN, each point on the convolution kernel learns an offset, which allows the convolution kernel to learn different structures in view of different parts of the data. This means that at each pixel point of the input feature map, we can learn a pair of offsets (x and y coordinates) that provide different convolutional effects for each location. These offsets are shared among different channels within the same feature map, thereby forming a deformable convolution module. The module structure is illustrated in Figure 1.

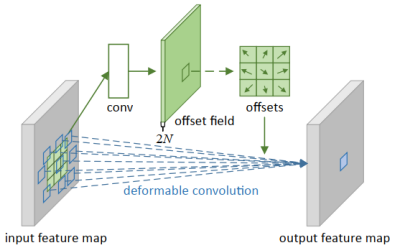


Figure 1. DCN module structure

Ordinary convolution is to sample a set of pixels from the enter feature map, and use

convolution operation to compute the sampling result to obtain the convolved result.

$$y(p_o) = \sum_{p_n \in R} w(p_n) \cdot x(p_o + p_n) \quad (1)$$

As for a deformable convolution, it is achieved indirectly by modifying the result of sampling to change the shape of the convolution kernel. Here we use Δp n to represent p n for expansion, where {Δpn | n = 1, 2, - - -, N}, in which case the deformable convolution is computed as:

$$y(p_o) = \sum_{p_o \in R} w(p_n) \cdot x(p_o + p_n + \Delta p_n) \quad (2)$$

The offset obtained by the convolution operation as above is a small number and cannot be sampled directly for use. So DCN uses the sampling method of bilinear interpolation to achieve the effect of using the offset, which is described by the above equation.

The proposal of transformable folding aims to improve the flexibility of convolutional neural networks to target objects with irregular shapes and expand its acceptance range. The traditional convolution operation is only suitable for feature extraction on regular rectangular receptive fields, while in true conditions, lots of target objects possess inordinance fashion, which requires the use of deformable convolution to recognize these irregularly shaped target substances. flexible convolution can adaptively adjust the form and extent of the receptive field according to the irregular shape of the target object, and this feature improves the robustness of the CNN in dealing with complex scenes. Therefore, we choose to add deformable convolution module to the backbone network of YOLOv8s. By adding the deformable convolution module, more and more detailed image features can be extracted, thus laying the foundation for feature fusion and prediction on the detection head later. It is used to improve the model's focus on small and medium-sized targets. After adding the deformable convolution module, because the sensory field of the location of the small target is adaptively changed, the model can better adjust the regression parameters of the prediction frame when the prediction frame is regressed, to increase the attention to the small target, and then improve the overall performance of the model.

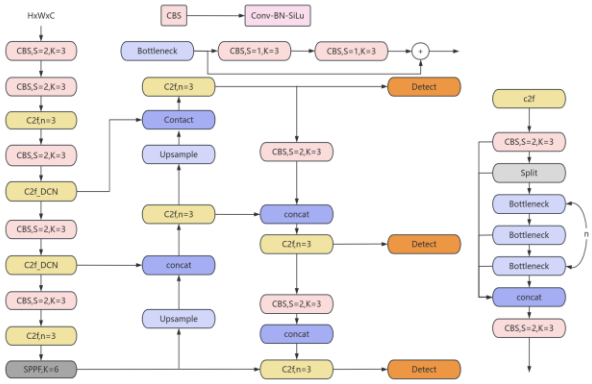


Figure 2. YOLOv8s+DCN

B. Attention mechanism CBAM improvement

In recent years, attention mechanisms have performed well in various deep learning tasks. Research has shown that attention mechanisms play a positive role in human visual perception, helping people efficiently and adaptively process visual information and focus on prominent areas of the image, thus enabling them to make the most accurate judgments. Among them, Convolutional Block Attention Module (CBAM) is a simple and effective attention module used to transmit convolutional neural networks. Our module leverages intermediate function cards to sequentially extract attention along two dimensions—channel and spatial. These attention cards are then multiplied with the input function cards to boost adaptability. As a lightweight and multifunctional module, CBAM can be easily integrated into any CNN architecture with minimal overhead and can be trained end-to-end alongside the CNN.

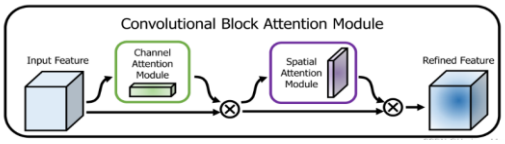


Figure 3. CBAM Network Architecture

There are several reasons to incorporate the CBAM attention module into the neck of YOLOv8s: (1) Enhanced feature representation: CBAM effectively adjusts channel and spatial weights in the feature map, allowing it to better

capture and represent key image features. (2) Improved computational efficiency: Compared to other attention modules, CBAM is less computationally demanding and more efficient (3).

Better task adaptability: CBAM is versatile and works well for a wide variety of visual tasks. The neck of the YOLOv8 network, situated between the backbone and prediction layers, plays a critical role in feature integration. Its structure allows for effective fusion of multi-scale features, which is crucial for accurate predictions. Thus, the design of the neck significantly impacts the algorithm's performance. In the modified network, as shown in Fig4, the CBAM module is placed after the un-sampling structure during the up-sampling phase of PAN-FPN and after each C2f module in the down-sampling phase, right before the CBS module convolution. This allows feature enhancement before fusion, enabling the model to focus more on small target details and improve the accuracy of both recognition and localization for small objects.

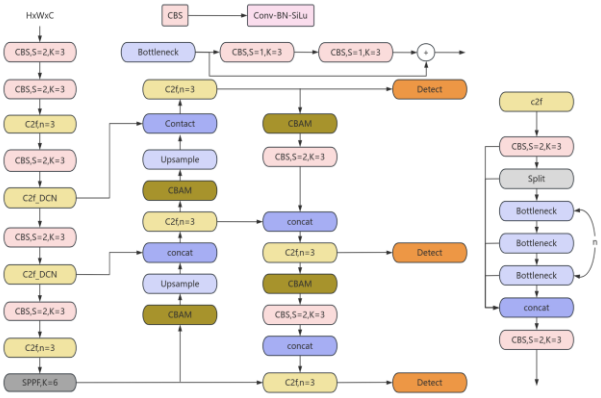


Figure 4. YOLOv8s+DCN+CBAM Network Architecture

C. Improvement of IOU

IoU Loss function is a loss function commonly used in target detection to measure the Intersection over Union (IU) ratio between the true and predicted frames. This loss function evaluates the accuracy of prediction by calculating the ratio of the intersection over union of two bounding boxes. Specifically, the size of the two boxes is first compared, the area of their overlapping parts is calculated, and then the area of the overlapping parts is divided by the total area of the two boxes to get the IoU Value. IoU The range of the value is [0,1], and a larger value indicates a higher similarity between the model prediction result and the real labeling. IoU The definition is as follows.

$$IOU = \frac{|A \cap B|}{|A \cup B|} \tag{3}$$

However, IoU has a big drawback. Firstly, if the two frames are not intersected, then IoU = 0, cannot reflect the size of the distance between the two. At the same time, because of the loss=0, there is no gradient back propagation, so we can't learn and train. Secondly, when the IoU ratio between the predicted and ground truth boxes is identical, the locations of the predicted boxes may still differ. Since the loss remains the same, this alone cannot determine which predicted box is more accurate.

Among them. $Wise-IoU$ A dynamic approach is used to compute the category prediction loss in theIoU loss, defined as follows.

$$L_{WIOU} = R_{WIOU} L_{IOU} \tag{4}$$

$$R_{WIOU} = \exp(\frac{(x-x_{gt})^2 + (y-y_{gt})^2}{(W_g^2 + H_g^2)^*}) \tag{5}$$

Among them. W_g , H_g , is the size of the smallest closed box. To prevent R_{WIoU} generating gradients that prevent convergence, W, H, is separated from the computational graph (superscript * indicates this operation), effectively eliminating the impediment to convergence.

In view of this, a new IoU criterion for loss function calculation, named GIoU is as follows:

$$L_{GIOU} = R_{WIOU} L_{IOU} + \frac{1}{2}(\frac{(x-x_{gt})^2 + (y-y_{gt})^2}{(W_g^2 + H_g^2)^*}) + \alpha v \tag{6}$$

The GIoU ability to dynamically adjust the bounding box regression loss is like that of the$Wise-IoU$. At the beginning of training, when the prediction box and the true box are IoU is small, the model should focus on boosting

IoU . Currently, the $R_{WIoU}L_{IoU}$ in the R_{WIoU} will be able to effectively increase the IoU smaller penalty strength. At the late stage of training, the prediction frame and the real frame are IoU is higher and stabilized. $R_{WIoU}L_{IoU}$ The value in L_{IoU} then decays to a smaller value, and the model automatically shifts the focus to the regression of the center point and aspect ratio to further accurately predict the box position, thus improving the performance. GIoU Improvements are made to the traditional IoU intersection and concatenation operations in the model. Simultaneously, dynamically adjust the bounding box regression loss and mitigate the penalty on geometric metrics such as distance and aspect ratio. This enables more comprehensive consideration of the differences between predicted boxes and actual boxes in terms of IoU, position, size, and shape, thereby improving the accuracy of object detection. By halving the second and third terms of the loss function, the GIoU intervene in the punishment of geometric metrics at a lower level, avoiding too much intervention in model training and enhancing the generalization ability of the model.

In a nutshell. GIoU In each scenario compared to CIoU and $\text{Wise} - \text{IoU}$ exhibits better adaptability and robustness to evaluate the target detection performance and classification tasks more effectively.

IV. EXPERIMENTATION

A. Introduction to the data set

The dataset is Safety-Helmet-Wearing, which contains 7581 images, including 6064 images in the training set and 1517 images in the validation set, and the dataset is labeled with Labelling tool. During the labeling process, the heads wearing helmets are labeled as "Helmet" and the heads not wearing helmets are labeled as "No Helmet".

Due to occlusion issues, the feature information in the image is weakened or disappeared. Therefore, the dataset annotation method is further optimized by reducing the detection contour of the safety helmet. The results are as follows:

Figure 5. Before

Figure 6. After

This improved annotation method will more accurately mark the key parts of the safety helmet, avoiding unnecessary background interference and greatly reducing the number of pixels required for annotation. This improved annotation method can mend the quality and stability of data.

B. Evaluation indicators

To evaluate the performance of the improved target detection model, the target detection model before and after the improvement is compared mainly in terms of detection accuracy and test speed. Firstly, taking the two types of categorized targets in this study as an example, when focusing only on all the images in the test set containing targets wearing helmets, the targets wearing helmets are counted as Positive cases (Positive), and the targets not wearing helmets are counted as Negative cases (Negative). The prediction results of the targets can be categorized into four cases: (1) True Positives (TP), which indicates lots of cases that were correctly classified as positive cases during the model detection process, in this paper, it is the number of instances of the target that actually wore a helmet and was classified as wearing a helmet by the YOLO model; (2) False Positives (FP), which indicates instances that were incorrectly classified as positive cases by the detected model. classified as positive instances, in this paper it is the number of instances where the target was not wearing a helmet but was classified by the YOLO model as wearing a helmet category; (3) True Negatives (TN), which indicates instances that were correctly classified by the detected model as negative instances, in this paper it is the number of instances where the target was

no-wearing a helmet and was classified by the YOLO model as wearing a helmet in the category of not wearing a helmet; and (4) False Negatives (FN), denoting the number of negative instances incorrectly classified by the detected model, in this paper the number of instances where the target was wearing a helmet but classified by the YOLO model in the category of not wearing a helmet.

Precision indicates the ratio of the number of correct samples detected to the total number of samples detected. In this paper, Precision refers to the proportion of images taken out as wearing helmet categories that are wearing helmets. It is calculated as Precision = TP/ (TP + FP)

Recall, also known as the check rate, indicates the ratio of the number of correct samples detected to the total number of samples in the test set. In this paper, it refers to how many images in the test set that contain the target wearing a helmet are correctly detected. A higher recall rate indicates better checking ability of the model. The specific formula for recall is as follows: Recall = TP/ (TP + FN)

Accuracy refers to the percentage of correctly detected samples in the total number of samples. Accuracy = TP+TN/ (TP +TN+FP+FN)

The curve formed by recall rate and accuracy, with recall rate as the horizontal axis and accuracy rate as the vertical axis, is called for the P-R curve. The area enclosed by the curve and coordinate axis is called the average accuracy, which is also an evaluation indicator for measuring the performance of the model on the dataset. mAP represents the mean of all categories of AP in the entire dataset in addition, under different threshold conditions, mAP will take different forms, among which mAP@0.5 It refers to the average accuracy of all categories in object detection tasks when IoU reaches 0.5. In common object detection evaluations, mAP@0.5 Usually used to evaluate the performance of algorithms.

C. Experimental results

During the training phase, we used tensor board to record the loss function of the model on the training and validation sets. As can be seen from the figure below, the training loss and validation loss of the model are gradually decreasing as the number of training times increases, indicating that the model keeps learning more accurate features. At the end of training, the model is evaluated on the dataset using the model and the following results are obtained.as shown in the figure7:

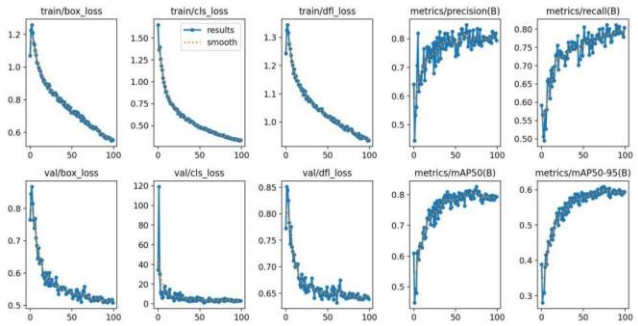


Figure 7. Loss Curve

Fig 8 illustrates the P-R curves of YOLOv8s, YOLOv8s-DCN, YOLOv8s-CBAM, YOLOv8s-DCN-CBAM, and YOLOv8s-improved models for each category on the Safety-Helmet-Wearing dataset.

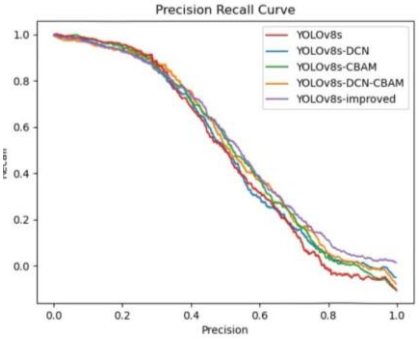


Figure 8. P-R curve

Table 1 shows the results of the ablation experiments on the dataset for different models:

TABLE I. EXPERIMENTAL RESULTS

Method				P (%)	R(%)	mAP@.5(%)
YOLO v8s	DCN	CBAM	GIOU			
√				67.3	58.9	64.6
√	√			72.6	56.3	64.3
√		√		71.4	60.1	65.7
√	√	√		70.1	58.4	65.7
√	√	√	√	72.5	59.3	65.9

As analyzed in figure4 P-R curve and Table 1, the YOLOv8s-improved algorithm is significantly more effective than other YOLOv8s-improved algorithms in the detection of safety helmets in terms of precision and recall. Adding CBAM attention module or DCN module separately to YOLOv8s can improve precision and recall respectively, but the overall detection performance is basically the same as that of YOLOv8s, which is not a great improvement to the overall performance of the model. From the last two rows of the table, adding the DCN module and the CBAM attention module to YOLOv8s, respectively, improves the accuracy rate while basically maintaining the recall rate, indicating that the added DCN module and CBAM module help to improve the network performance. Finally, the new IoU loss function calculation criterion is applied to the yolov8-DCN-CBAM model, and compared with YOLOv8s, YOLOv8s-improved can effectively improve the target miss detection while guaranteeing the recall rate.

V. Conclusions

Using YOLOv8 as the basic model to realize the worker's helmet in complex scenes detection can timely detect and correct unsafe behaviors and effectively reduce number of incidents, minimizing casualties and economic losses. The purpose of this paper is to construct a helmet detection algorithm, which innovatively incorporates the improved YOLOv8s model and proposes the YOLOv8s-improved algorithm. The key to this improvement is to add realizable convolutional modules in the backbone network to improve the perceptual field of the points detected on the feature map; such a structural improvement injects the whole algorithm with a deeper level of information comprehension and improves the perceptual power of the model, which in turn improves the detection of workers wearing helmets. Increasing the attention mechanism in the neck to focus more on local details and edge information retention, which enables the model to better capture the target's textures, boundaries, and small changes, which significantly improves the detection and localization of small targets and more effectively reduces the occurrence of missed detection. IoU Guidelines allow for Designing Networks makes it possible to adaptively adjust the proportions of each part of the loss function at different stages to better address the detection challenges in different scenarios. The experiments show that the improved YOLOv8s-improved model has improved precision in detection, increased recall, reduced missed detections, and the detection speed meets the requirements of real-time detection, which provides a certain means for the subsequent detection of factory workers' safety helmets in complex scenarios.

References

[1] Hui Wang. Helmet Detection and Identity Recognition Based on Improved Faster R-CNN. Xi'an University of Science and Technology, 2020.

[2] Chun hui Xie, JinMing Wu, Haiyu Xu. Improved small target detection algorithm for UAV imagery with YOLOv5. Computer Engineering and Applications, 2023, 59(9): 198-206. XIE C H, WU J M, XU H Y. Small object detection algorithm based on improved YOLOv5 in UAV image. Computer Engineering and Applications, 2023, 59(9): 198-206.

[3] Xu K, Deng C. Helmet wearing recognition algorithm based on improved yolov3. Progress in laser and Optoelectronics, 2021, 58(06):300-307.

[4] Jin F Y, Wu X, Dong H, et al. Helmet detection algorithm based on improved yolov4. Computer science, 2021, 48(11): 268-275.

[5] XiaoFeng SONG, YunJun WU, BingBing LIU, et al. Improved YOLOv5s algorithm for helmet wearing detection. Computer Engineering and Applications, 2023, 59(2). 194-201. Song X F, Wu Y J, Liu B B, et al. Improved YOLOv5s algorithm for helmet wearing detection. Computer Engineering and Applications, 2023, 59(2): 194-201.

[6] Redmon J, Divvala K S, Girshick B R, et al. You Only Look Once: Unified, Real-Time Object Detection. CoRR, 2015, abs/1506.02640.

[7] Redmon J, Farhadi A. YOLO9000: Better, Faster, Stronger. CoRR, 2016, abs/1612.08242.

[8] Yifang Chen, Shang Zhang, Xiukang Ran, et al. Improved YOLOv8-based aircraft target detection algorithm for SAR images. Telecommunication Technology, 2023-08-04.

[9] Huantong Geng, Zhenyu Liu, Jun Jiang, et al. An embedded road crack detection algorithm based on improved YOLOv8. Computer Application, 2023-08-28.

[10] DAI J, QI H, XIONG Y, et al. Deformable convolutional networks. Proceedings of the IEEE international conference on computer vision. 2017:7

Long-term Target Tracking Based on Template Updating and Redetection

Shuping Xu
School of Computer Science and Engineering
Xi'an Technological University
Xi'an, China
E-mail: 563937848@qq.com

Yinglong Li
School of Computer Science and Engineering
Xi'an Technological University
Xi'an, China
E-mail: 373301967@qq.com

***Abstract*—To address the issue of targets frequently disappearing and reappearing in long-term tracking scenarios due to occlusion and being out of view, we have developed a long-term target tracking algorithm based on template updating and redetection (LTUSiam). Firstly, on the basis of the basic tracker SiamRPN, a three-level cascade gated cycle unit is introduced to assess the state of the target and select the right time to adopt the template update network to adapt the update template information. Secondly, a re-detection algorithm based on template matching is proposed. The candidate region extraction module is utilized to adjust the target's position and size in the basic tracker, and the evaluation score sequence is used to judge the target loss to determine the tracking state of the next frame. Experiments show that LTUSiam achieves 28 frames per second on VOT2018_LT dataset, achieving good results in real-time tracking, and 0.644 performance on F-score, which has better robustness in handling the problem of target loss recurrence, and effectively improves the performance of long-term tracking.**

Keywords-Long Term Tracking; Twin Network; Template Update; Reinspect

I. INTRODUCTION (HEADING 1)

Target tracking involves using size and position information of the target from the initial frame to estimate its location in subsequent frames. Visual target tracking has applications in various fields [1-3], including autonomous driving, robotics, safety, and surveillance. Based on the length of the sequence, tracking tasks are divided into short-time tracking and long-time tracking. At present, many algorithms mainly study the short-time tracking, which mainly solves the tracking challenge that the target is always visible and the video frame is short. However, long-term tracking is more aligned with the highly challenging real-world scenarios, in the task may need to continue to track the target for several minutes or even hours, and there are frequent target disappearing and reappearing, so the study of long-term tracking is of great practical significance.

At the beginning, the appearance model of long-term tracking used manual features to describe the target, but the use of manual features resulted in weak feature representation of the target, which could not cope with the challenges of complex scenes. However, the emergence of deep learning alleviated the problem [4-6] of inadequate feature representation to a certain extent. Zhang et al. proposed an MBMD algorithm combining regression network and validation network to dynamically switch the search mode through online learning of a classifier, and identify the redetection within the whole graph by using a sliding window after the target is lost. However, the direct sliding window strategy and online learning verification module made the model run very slowly. Which is far from real time applications [7]. Zhu et al. propose a long-duration tracking algorithm, Dasiam_LT, which enhances the original tracker by incorporating a strategy that transitions from a local to a global search region. The distraction-aware module is used for training and inference to determine whether the tracker fails to track, and iteratively increases the size [8] of the search area when the tracking fails. The Dasiam_LT tracker has demonstrated commendable performance in the long-term challenges of VOT2018; however, it necessitates a substantial amount of image sequences for offline

training. Dai et al proposed LTMU algorithm, which uses off- line training meta-updater for online tracking, and introduces validation network into short-term tracker, so that long-term tracking can improve performance on the basis of short-term target tracking algorithm [9]. Huang et al proposed a GlobalTrack algorithm founded on global instance retrieval, built a target-specific object detector founded on Faster R-CNN, utilizing a convolution module to learn how to adjust the characteristics of the search region by leveraging the target template's region of interest [10]. While this algorithm enhances accuracy, its real-time performance is lacking, and it fails to locate the target when it is too small.

To address the aforementioned issues, this paper will improve SiamRPN network and propose a long-term target tracking algorithm (LTUSiam) grounded on template updating and redetection. Specific contributions include: (1) a redetection algorithm based on loss judgment mechanism is proposed, which combines the initial target template with the confidence score to judge the disappearance of the target. When the target is lost, the redetection algorithm based on template matching is used for relocation. (2) A state-based template updater is introduced, consisting of two components: the status judgment module and the template update module. The status judgment module primarily addresses the timing of updates, while the template update module focuses on the method of updating. (3) LaSOT [11] datasets demonstrate that the proposed method exhibits strong performance.

II. LONG-TERM TARGET TRACKING BASED ON TEMPLATE UPDATING AND REDETECTION

The overall architecture of the algorithm is illustrated in Figure1. In each frame, the SiamRPN algorithm is used as the base tracker, and the SiamRPN tracker is used for local search, the bounding box and similarity score of the tracked target are output. Then, the accuracy of the current tracking result is evaluated through the loss judgment mechanism. If the tracking result is accurate and the target is not lost, local tracking is still carried out in the next frame. If the tracking result is not accurate, the target is judged to be lost, and the redetection algorithm is used to search the global image.

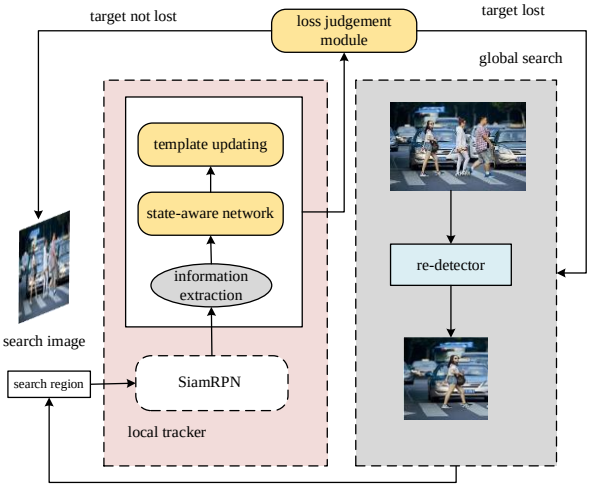


Figure 1. Block diagram of long-term target tracking algorithm

Based on the SiamRPN algorithm, the short-term local tracker uses the SiamRPN tracker to conduct local search firstly, obtain the bounding box position size and similarity score of the target, and judge the disappearance of the target through the evaluation score, and then determine the tracking strategy for the next step. In the local tracker module, an adaptive template updating mechanism is introduced to mitigate noise interference, ensuring that the optimal template is updated at appropriate intervals. This approach addresses the challenges of deformation in long-term tracking scenarios and enhances the accuracy of the local tracker. The evaluation score sequence is employed to assess the potential disappearance of the target. If the target is deemed lost, a global instance search is conducted using a template matching redetection algorithm, after which the bounding box with the highest classification score is selected as the target's reappearance location.

A. Local tracker based on adaptive template update

During the long-term target tracking, the accuracy and robustness of the local tracker are crucial to the tracking results. In real-world complex scenarios, the target frequently becomes lost, further complicating the tracking process, the target's reappearance also impacts the tracker's performance. In this chapter, SiamRPN algorithm

is used as the local tracker. To tackle the challenge of target deformation, template updating mechanism is introduced. However, template updating is a double - edged sword in terms of noise introduction and information description. For long-term tracking, if the template is updated at an inappropriate time, there will be long-term cumulative errors and inappropriate samples collected, which may result in model degradation and tracking drift. Based on this, a template updater based on state judgment is put forward to address the issue of when and how to update, and then update the target template in a robust manner. Figure2. shows the detailed framework of the template updater based on state judgment, which comprises two principal components: the state judgment module and the template update module.

1) Status judgment module

In the state judgment module, the geometric features, appearance features and discrimination features are integrated according to the time sequence information, and the sequence matrix is input into the three- level cascade gated cycle unit. Ultimately, the two fully connected layers are employed to evaluate the reliability of the current tracking state, specifically determining whether the template should be updated in the present frame. The state judgment module mainly consists of two parts: information extraction and state awareness network.

a) Information extraction.

In the basic local tracker part, the geometric features, appearance features and discrimination features of the local tracker in the current frame are mined, and then the sequence matrix is formed by combining the timing information in the previous frame within a given period of time, which is used as the input information of the state-aware network.

Geometric features that describe the location and size of a target. The target tracking algorithm SiamRPN will output a four-dimensional vector every frame, which can be used to calculate the position information of the boundary box. In the t frame, the bounding box $b_t = [x_t, y_t, w_t, h_t]$ obtained from the tracker is used as the tracking result, where (x_t, y_t) represents the top-left corner coordinates of the target and (w_t, h_t) represent the target's width and height, respectively.

As can be seen from the coordinate information of the bounding box, it can only provide the geometric position data of the target being tracked in the frame at this moment. Nevertheless, in the target tracking task, it is usually necessary to model the motion state of the target. Since the position, shape and size of the target fluctuate between successive frames, the motion state of the target can be estimated by comparing the boundary frame information between successive frames, and then the speed and acceleration of the target can be obtained. It is easier to capture the motion mode of the target and improv e the robustness and accuracy of the tracker by describing b_t in the upper left corner and upper right corner $(x_t^1, y_t^1, x_t^2, y_t^2)$

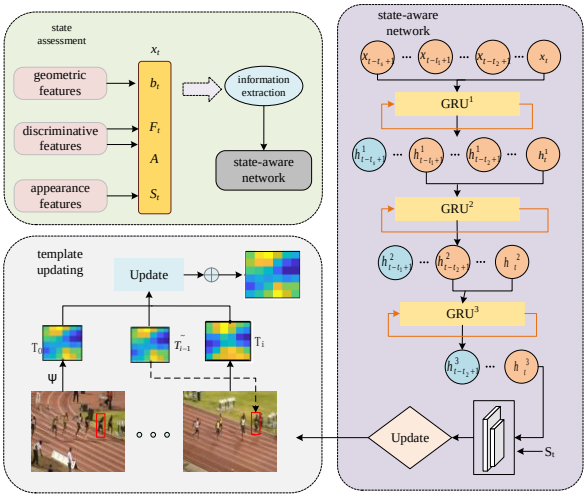


Figure 2. Template updater based on state judgment

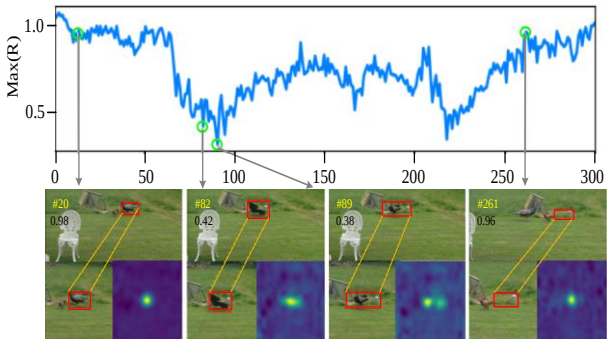


Figure 3. Confidence score chart

Discriminant features, used to differentiate the target from surrounding background information. The SiamRPN algorithm finally outputs a feature response graph R_t whose maximum response value can be used to represent the confidence score of the bounding box b_t as shown in formula (1).

$$F_t = \max(R_t) \tag{1}$$

Figure3. shows the confidence scores in the tracking process. The results show that the confidence scores of frames 89 and 261 are unstable, so the quality assessment value is used for auxiliary discrimination and the discrimination information in the response graph is thoroughly mined. The calculation formula is shown in Equation (2)

$$A = \mu_1 \frac{\left|F_{\max} - mean(F_{\max})\right|}{mean(F_{\max})} + \mu_2 \frac{\left|apce - mean(apce)\right|}{mean(apce)} \tag{2}$$

Among them, A represents the quality evaluation value, $F_{\max}$ represents the highest response data on the response map, $apce$ represents the average peak correlation quantity, The formula for its calculation is provided in the following equation (2.3)

$$\text{apce} = \frac{\left|F_{\max} - F_{\min}\right|^2}{mean(\sum_{x,y}(F_{x,y} - F_{\min})^2)} \tag{3}$$

Among them, $F_{\min}$ represents the minimum value of the response graph, $F(x,y)$ represents the response value associated with the coordinates (x,y).

Appearance feature, which is utilized to indicate the similarity between the appearance of the target template and the current frame target. Using noise samples for template updating usually makes the response graph insensitive to appearance changes, so the method of template matching can be used as an important supplement and similarity score can be defined, as shown in formula (4).

$$S_t = \cos(I_t, I_0) \tag{4}$$

Where I_0 represents the initial template feature and I_t represents the tracking result of the current frame. Timing information: geometric features, discriminant features and appearance features are combined into column vectors X_t, as shown in formula (5)

$$X_t = [x_{t-t_s+1}, \ldots, x_{t-1}, x_t] \tag{5}$$

Where t_s is the time step utilized to balance historical and current information, so that the temporal information X_t includes both the motion and appearance changes of the target. The temporal information is then fed into the state-aware network to judge the target state and decide whether to update the target template information.

b) State aware network.

It mainly uses the timing information to judge whether the current frame needs template updating. The input data is a sequence matrix, so it can be processed by recurrent neural network (RNN). However, RNNs may encounter the issue of gradient vanishing when addressing long-term dependencies. The gated cycle unit (GRU), a variant of recurrent neural network, can reduce the problem of gradient disappearance through the gating mechanism while retaining more long-term sequence information. at the same time, the training speed is faster and the effect is better, so the GRU network model is selected for this module to process the input long-term sequence data. The model incorporates two gating mechanisms: reset gate r_t and update gate z_t .By filtering and updating the historical information and the current input, the network model can better process the sequence data.

The update gate is used to control the residual amount of previous data retained to the current moment. The smaller the value, the less historical information is retained. Its mathematical description is shown in Equation (6)

$$z_t = \sigma(W_z x_t + U_z h_{t-1}) \tag{6}$$

Where h_{t-1} represents the hidden state of the previous moment, W_z and U_z represents the weigh information, x_t refers to the input at the present time, σ denotes the activation function of $Sigmoid$. It is mainly used to normalize data and can act as a gating signal.

The reset gate governs the extent of information that should be discarded from the previous moment. The smaller the output value, the more information needs to be discarded and ignored. The specific mathematical description is shown in Formula (7).

$$r_t = \sigma(W_r x_t + U_r h_{t-1}) \tag{7}$$

The mathematical representation of the hidden layer's state at the current moment is presented in the following equation (8).

$$h_t = (1 - z_t) \cdot h_{t-1} + z_t \cdot \hat{h}_t \tag{8}$$

Where $\hat{h}_t$ the mathematical description of the candidate state is shown in equation (9).

$$\hat{h}_t = \tanh(wx_t + u(r_t \cdot h_{t-1})) \tag{9}$$

Where is the Hadamard product of the matrix.

The sequence matrix obtained from the information extraction part is input into the gated cycle unit of the three - level cascade for calculation and analysis. Simultaneously, to further strengthen the appearance features, the output h_t^3 obtained through the gated cycle unit of the three- level cascade is connected with the appearance features (S_t), and then the two vectors are processed through two fully connected layers to produce a binary classification fraction. Employed to determine whether the template should be updated in the current state.

2) Template update module

Most trackers use linear interpolation or a straightforward average weighting strategy, as illustrated in formula (10), to update the template.

$$\tilde{T}_i = (1 - \partial) T_{i-1} + \partial T_i \tag{10}$$

Where i denotes the number of frames of the video sequence, T_i represents the new template derived from the frame at this moment, $\tilde{T}_i$ signifies the cumulative template, η is the update rate, set to a fixed value of 0.01.

However, there are two problems in using the simple weighted average strategy: (1) The update rate is a constant value, leading to a somewhat simplistic update mechanism; (2) No initial template frame information is used, which easily leads to tracking drift. Based on this, this excerpt uses a generic function φ derived from adaptive update template features, where the function φ is implemented using the UpdateNet network model, which is capable of learning from extensive datasets. The new template information is derived by integrating the initial template frame T_0, the previously accumulated template frame $\tilde{T}_{i-1}$, and the template of the target position estimated by the frame at this moment T_i,as shown in equation (11).

$$\tilde{T}_i = \varphi(T_0, \tilde{T}_{i-1}, T_i) + T_0 \tag{11}$$

Figure2. Gray dashed line box describes the specific structure and overall framework of the template update module, using the feature extraction network proposed in Chapter 3 to extract the feature information of the target from the image. During the course of template update, the information of the first frame is real and reliable, so the template features T_0 can be extracted at the target boundary box position given in the initial frame. To get T_i, we first need $\tilde{T}_{i-1}$ to ascertain determine the position of the target of the frame at this moment, and then use the feature extractor to extract the feature information T_i of the current frame within this region. The input to UpdateNet is a triplet of T_0 (leftmostfeature map, initial template feature), $\tilde{T}_{i-1}$ (dashed line connection, previous frame accumul ated template

feature) and T_i (rightmost feature map, current frame template feature). In the first frame, since there are no previous frames, so $\tilde{T}_{i-1}$, T_i and T_0 is initialized. Of the three inputs to UpdateNet, Only the information from the initial frame is true and reliable, while the information in subsequent frames is predicted by the tracking algorithm, the rest is predicted by the tracking algorithm, so T_0 can be used as a reliable signal to guide the model update. Based on this, skip connections are used to combine the initial template feature f0 with the output of UpdateNet, to achieve the most accurate template features.

3) The re-detection algorithm based on the missing judgment mechanism

4) Target loss judgment mechanism

Figure 4. shows the Jogging effect of SiamRPN algorithm in OTB2015 dataset. The top graph represents the confidence score you get when you track a video sequence with SiamRPN. The following picture illustrates the tracking results of SiamRPN algorithm across various frames. The red bounding box indicates the tracking output of SiamRPN algorithm, while the black bounding box indicates the actual location of the target. From the figure, it can be seen that among the initial frames of the sequence, two girls are Jogging into the field of vision, and the girl wearing black pants is the target being tracked. The target object is always moving from the initial frame to the 39th frame, and SiamRPN algorithm keeps tracking it accurately. However, between the 39th frame and the 73th frame, a telegraph pole appears, and the target object is completely covered and disappears into the field of view during this period. simultaneously, the confidence score decreases sharply. When the target is lost because of occlusion and other factors, the confidence score will also be reduced, so the confidence score can be used to judge the disappearance of the target. However, due to the integration of the adaptive template update mechanism in the tracker, the confidence score does not decrease significantly. Therefore, this section uses the initial template features obtained in the first frame to make further judgment on the basis of the confidence score obtained by the algorit.

Firstly, the Euclidean distance between the target initial template z and the tracking result x predicted by the algorithm is calculated as the similarity, and the formula is shown in equation (12)

$$D = \left\| z - x \right\|_2 \tag{12}$$

Then the similarity score D is combined with the confidence score s to judge the disappearance of the target, the formula is presented as follows.

$$r = mean(D + s) \tag{13}$$

Defined r as evaluation score, the evaluation score of consecutive video frames is utilized to assess the disappearance of the target.

During the tracking process, the target may be lost only a few frames or the algorithm itself has calculation errors, so the delay judgment is also needed to ensure the compatibility of the algorithm and the stability of the tracking. The detailed flowchart is presented in Figure 5. In general, when the evaluation score r falls below the specified threshold t, the cumulative number of failures is set to 0 when the evaluation score is greater than the given threshold. When the evaluation score t is less than and the number of failures f or better than the loss threshold c, the target is considered lost and the number f of failures is set to 0.

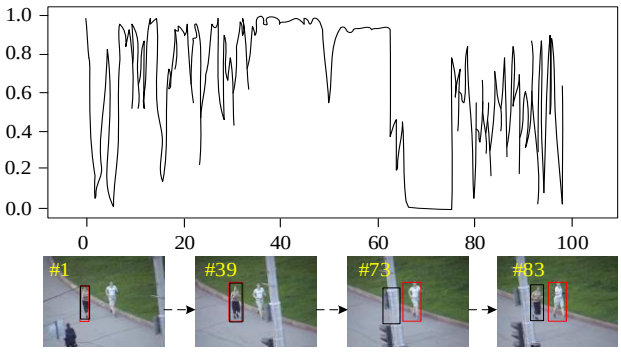


Figure 4. SiamRPN tracking results

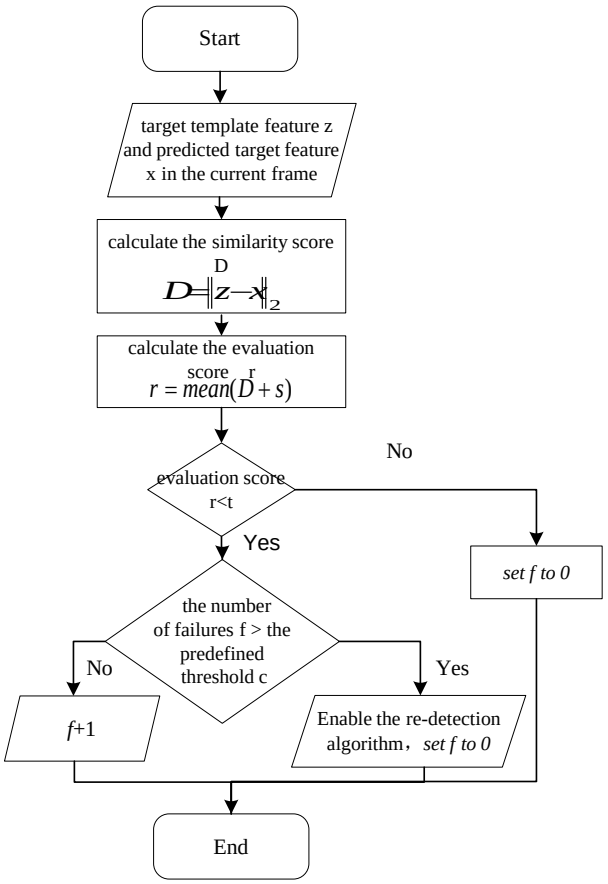


Figure 5. Flow chart of target loss judgment mechanism

5) *Redetection algorithm based on template matching*

When the local tracker identifies that the target has been lost through the target loss judgment mechanism, it must initiate a global search to redetect the target within the subsequent frame's image area and identify the most likely location of the tracked target. Consequently, the redetection algorithm must swiftly scan the entire image and accurately pinpoint the target's location without relying on historical frame information. Based on this, a redetection algorithm based on template matching is proposed.

As shown in Figure 6, the redetection algorithm based on template matching mainly is primarily composed of three components, namely, feature extraction module, candidate frame extraction module and precise positioning module. To enhance the redetection algorithm's ability to differentiate between the background and the target amidst similar interference, a cross-query loss function is employed to optimize the algorithm.

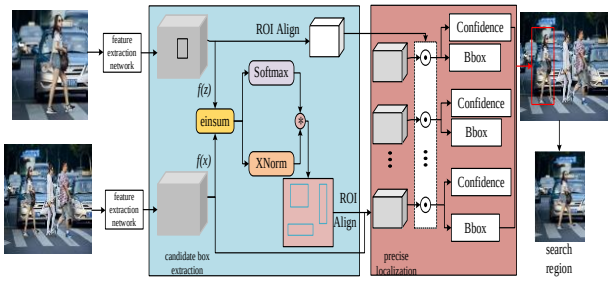


Figure 6. Template-based re-detection algorithm

a) Feature extraction.

The feature extraction network built on feature pyramid is used to extract template frame and search frame feature. In the global search, compared with the whole image, the tracked object can be regarded as a small target. The feature pyramid network can extract the deep semantic information and shallow detail information of the target to the maximum extent, and improve the ability of the heavy detector to locate small targets.

b) Selection of candidate box.

Through the feature extraction module, feature extraction is performed on the template map z and search map x, and the feature map $f(x)$, $f(z)$ is output. Simultaneously, to enhance the target's feature information, the feature map s is generated by summing up $f(x)$ and $f(z)$ using einsum, and the calculation process is detailed in Formula (14).

$$s(x,z)=\sum_{i=1}^{c} f(x)f(z) \tag{14}$$

Where c denotes the number of feature channels.

Subsequently, a $Soft\max$ function is used on the feature map to calculate the probability that each location may contain the target region, as shown in equation (15).

$$p=Soft\max(s(x,z)) \tag{15}$$

When the background of the image is too complex, it is impossible to obtain accurate information only by sampling the feature map

using $Soft\max$ functions. Therefore, *XNorm* constraints can be used to assist the discrimination of the feature information, obtain the weight matrix that highlights the effective information, and then multiply the generated probability matrix and weight matrix to generate the fraction matrix after enhancing the effective information, as shown in (16).

$$\begin{cases} w = \dfrac{s(f(x)) \cdot s(f(z))}{\sqrt{\sum_{i=1}^{c} \left| s(f(x)) \right|^2}} \\ r = w \cdot p \end{cases} \tag{16}$$

Then using the maximum value calculated by the $\arg\max$ function, set the candidate box anchor point on that location region to generate a series of candidate regions. The loss function is the same as RPN.

c) Precise positioning module.

Primarily tasked with the categorization and regression of the candidate region generated by the candidate box extraction module. Firstly, execute the ROI Align operation on the target template and various candidate boxes generated by the candidate region extraction module to get the ROI characteristics of the target template and candidate region; Then, assess the similarity between the two ROI features, the specific formula is shown in equation (17).

$$\tilde{x} = h_s(h_x(x_i) \cdot h_z(z)) \tag{17}$$

Among them, x_i represents the ROI feature of the candidate box, $\cdot$ represents the Hadamard product, z represents the ROI feature of the target template, h_s represents using a 1x1 convolutional kernel to change the number of channels in a tensor, h_x and h_z represents the convolution operation, the dimensions of the convolution kernel is 3×3 and the fill is 1. Then, the traditional RCNN method is used to conduct target classification and boundary box regression for the feature maps $\tilde{x}$ obtained by similarity coding.

III. Experimental results and analysis

A. Preparation of experiment

The experiments in this chapter are completed on a PC using Pytorch deep learning framework, GPU is GetForce RTX 2080Ti, memory size is 64G, the algorithm in this chapter is written based on python language.

a) Template update module.

The template update module is trained using a three-stage training method. First, the training of the first stage obtained he cumulative template $\tilde{T}_i$ by a simple average weighting strategy. The calculation formula is shown in equation (18).

$$\tilde{T}_i = (1-\eta) \cdot \tilde{T}_{i-1} + \eta T_i \tag{18}$$

Among them, T_i denotes a new template calculated in the first stage of training using the current frame, parameter $\eta = 0.01$. Secondly, during the second and third stages of training, the cumulative template is derived by updating the module with the adaptive template proposed in this chapter, and the weight data is obtained by using the parameters trained in the previous stage. The LaSOT dataset is a seminal resource in the realm of long-term target tracking, characterized by its complex and varied video sequences. However, the template update module only contains two layers of convolutional neural networks, so the update module with 20 video sequence training templates is sufficient to meet the requirements.

In the first stage of training, set the starting learning rate as 10^{-6}, and with the weights initialized randomly. After each epoch is trained, the learning rate will be logarithmically decayed; In the second stage of training, the parameters of the optimal model obtained from the first stage are utilized to initialize the weights, and the learning rate is attenuated from $10^{-7}, 10^{-8}, 10^{-9}$ to $10^{-9}, 10^{-10}, 10^{-11}$; The third stage imports the optimal model from the second stage, and the learning rate is attenuated from $10^{-8}, 10^{-9}, 10^{-10}$ to $10^{-9}, 10^{-10}, 10^{-11}$. The stochastic gradient descent algorithm is selected to train the template update module, in which the

weight attenuation and momentum are set to 0.0005 and 0.9 respectively.

b) Heavy detector.

The COCO data set is used to train the redetection module, and data enhancement techniques are used to generate more image samples [12] in the pre-processing stage. The model was trained 50 times in total. The average loss of candidate region extraction module and precision positioning module was used as the total loss function, and the SGD method with momentum of 0.9 was used to optimize [13] the network model.

B. Parameter analysis

a) State judgment module

The size of the time step t_s of the status judgment module is crucial to the tracking results, t_s including the information of the present and the historical frame, its value determines the richness of the obtained timing information. The success rate and precision of the short-term local tracker based on the state judgment module were calculated on the OTB2015 dataset for different values of t_s, and the experimental results are shown in Figure7. Among them, the horizontal coordinate t_s represents the size, the left and right vertical coordinates represent the success rate and accuracy rate, Z are represented by red and green curves respectively. As illustrated in the figure, when $t_s = 25$ both the success rate and accuracy rate of the local tracker on the OTB2015 dataset have achieved the maximum value, and the tracking performance is effective at this time.

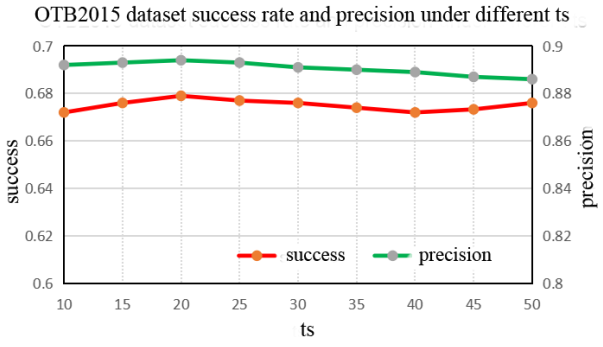


Figure 7. Different t_s corresponding success rates and accuracy

b) Target loss module

In the experimental verification of the target loss judgment module, use μ as the evaluation score threshold to judge whether to switch from local tracking to global tracking, use δ as the number of lost, and then test it on the OTB2015 dataset, use the sum of accuracy and success rate as the judgment standard, and select the appropriate threshold. Figure 8 is the performance score chart under different values δ and values μ, where Y axis is the evaluation score threshold, Z axis is the sum of the benchmark success rate and accuracy rate, and X axis represents the number of lost times. In the experiment, the value μ is 1 to 12, μ from 0.08 to 0.2, with the constant change of μ、δ, it can be seen from the figure that when $\delta = 10$, $\mu = 0.13$, the tracking performance is the best.

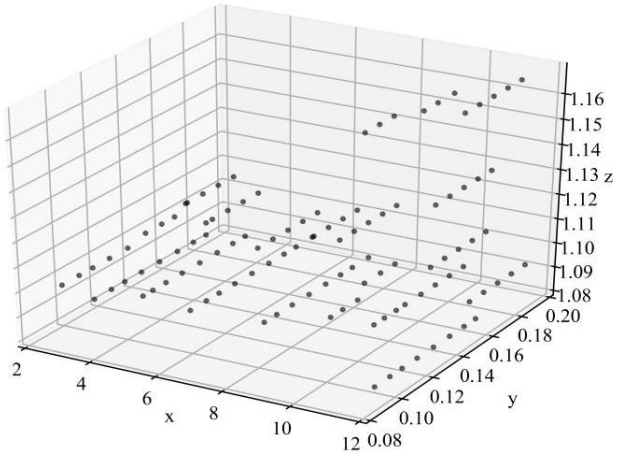


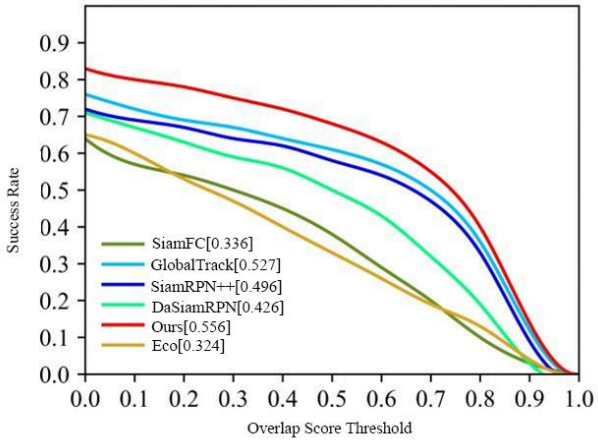
Figure 8. Results of parameter optimization

C. Quantitative experimental analysis

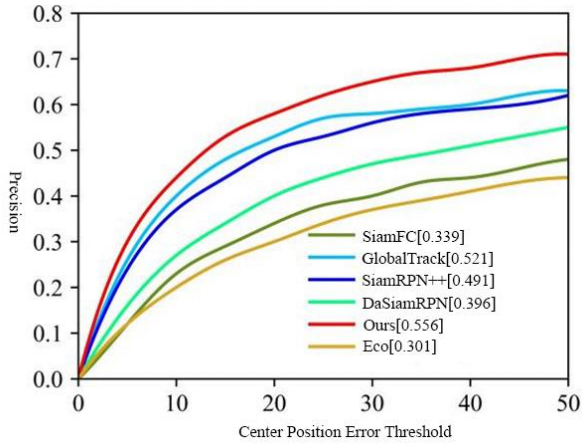
Conduct quantitative experimental analysis of LTUSiam algorithm with other existing advanced trackers on LaSOT and VOT2018_LT datasets.

For LaSOT test set, LTUSiam algorithm is compared with five tracking algorithms, namely SiamFC [13], GlobalTrack, SiamRPN++ [14], DASiamRPN and ECO [15]. As shown in the figure, LTUSiam algorithm has the best tracking effect on LaSOT test dataset, with the success rate and accuracy rate reaching 0.566 and 0.556 respectively, which indicates that LTUSiam, the improved algorithm in this chapter, can effectively handle the target loss recurrence scenario. At the

same time, LTUSiam algorithm can achieve a tracking speed of 25 f/s on LaSOT data set, meeting the real-time tracking requirement.



(a) success rate



(b) Accuracy

Figure 9. Diagram of LaSOT experimental results

The LTUSiam algorithm is compared with other 5 tracking algorithms SiamFC, SPLT, SiamRPN++, DASiamRPN_LT and MBMD, and the experimental results on the VOT2018_LT dataset are presented in Table 3.1. VOT2018_LT dataset uses precision rate (P) and recall rate (R) as evaluation indexes [16-17]. When there is a contradiction between P and R (for example, P value is high but R value is low), the results of precision rate and recall rate are comprehensively considered, and F value is used as evaluation index. The higher F value is the better tracking performance is. As indicated in the table, although the algorithm discussed in this chapter is in the middle position in terms of frame rate of 28fps, it has achieved relatively good results in terms of accuracy, accuracy, recall rate and F-value. Experiments show that LTUSiam algorithm has good tracking performance and fast speed in long-term sequences.

TABLE I. EXPERIMENTAL RESULTS ON VOT2018_LT

Algorithm	*F-value*	*Accuracy*	*Frame rate*	*Recall rate*
SiamFC	0.429	0.628	84	0.323
MBMD	0.613	0.636	4	0.576
DASiamRPN_LT	0.604	0.625	63	0.585
SPLT	0.614	0.629	26	0.602
SiamRPN++	0.625	0.646	35	0.606
Ours	0.644	0.659	28	0.626

D. Qualitative experimental analysis

To more directly assess the tracking performance of the LTUSiam algorithm, two representative video sequences were selected from the VOT2018_LT dataset for analysis. The results were compared with those of several leading tracking algorithms, including SiamFC, SPLT, SiamRPN++, DASiamRPN_LT, and MBMD. Figures 10 and 11 illustrate the tracking outcomes of seven different trackers under challenging conditions such as deformation, vanishing and reappearance, and occlusion. Video sequences with challenge factors such as target recurrence and deformation are mainly selected for visual analysis, and their specific introduction is shown in Table 3.2.

TABLE II. INTRODUCTION OF 2 GROUPS OF VIDEO SEQUENCES

Video Sources	*Name*	*Number of frames*	*Type of challenge*
VOT2018_LT	Yamaha	3143	Out of sight, occlusion, deformation
VOT2018_LT	bird1	2437	Analogue interference, out of view, blocking

As shown in FIG10. In the bird1 video sequence, the tracked object bird has problems such as long time out of field of view and deformation. From frame 1 to frame 22, the bird needs to stir its wings during flight, resulting in drastic changes in the shape of the target, and algorithms such as SiamFC and DaSiamRPN_LT cannot adapt to the changes in the appearance of the target, leading to tracking failure. The adaptive template updating mechanism of LTUSiam

algorithm selectively updates the template through the state judgment module. So that the algorithm can track the target stably; from frame 196 to 219, the bird experienced partial and full occlusion when flying over the wire. SiamFC and SiamRPN algorithms could not fully extract the feature data of the object, resulting in tracking drift; From frame 259 to frame 520, due to the camera's restricted field of view, the bird flew out of the target area and did not appear for a long time, LTUSiam algorithm and MBMD algorithm have been stably tracking the target in this process, and other algorithms cannot cope with the disappearance of the target due to the lack of redetection module. And not relocating to the target area properly after the object reappeared. Therefore, our algorithm can solve the problem of disappearing and reappearing in the tracking process.

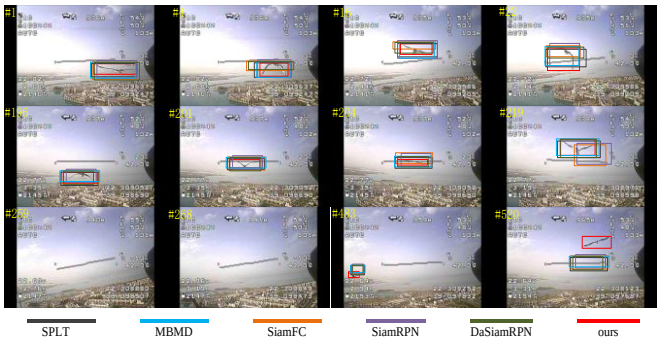


Figure 10. Results of qualitative analysis of bird1 video sequence

As shown in Figure11, in the yamaha video sequence, cameras fail to capture or only capture part of the target many.

Times during the movement of the tracked motorcycle. In addition, in order to maintain the balance of the body, the motorcycle needs to tilt at a certain Angle when turning, which leads to the frequent disappearance, recurrence, deformation and other problems of the target object during the operation. As can be seen from the figure, from the first frame to the 150th frame, the target object runs normally, and the algorithm tracks the target stably and accurately; From the 167th frame, the motorcycle tilts at a certain Angle, but the Angle is small, so the deformation is not obvious, all algorithms still track the target, but to the 217th frame, the motorcycle deformation is obvious, some algorithms cannot adapt to the change of scale, and the tracking results drift; By frame 272, only MBMD and LTUSiam have been tracking the motorcycle stably. From the 492th frame, the target gradually disappeared from view, to the 507th frame, the target completely disappeared, until the 522th frame again, in this process, only the algorithm in this chapter can re-search the target through the redetection algorithm after the target is lost and reappeared, and stable tracking. Starting from the 2539 frame, part of the position of the motorcycle was obscured, and only part of the target could be seen. Our algorithm could quickly locate the position of the motorcycle for accurate tracking.

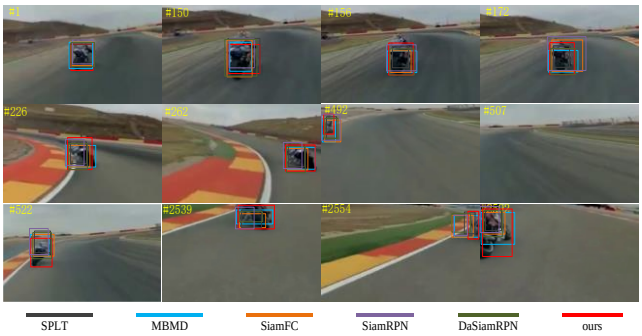


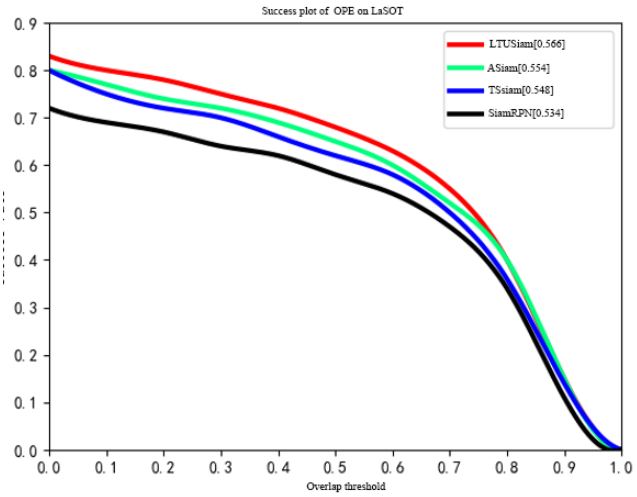
Figure 11. yamaha video sequence qualitative analysis results

E. Ablation Experiment

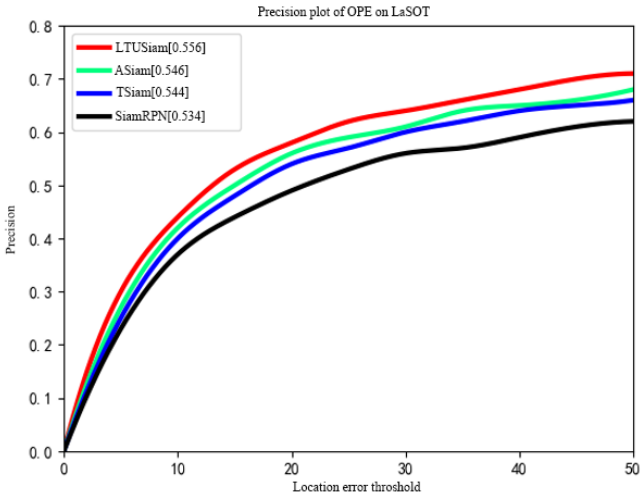
For the purpose of more fully demonstrate the vaildness of the long-term target tracking framework, adaptive template tracking strategy and global redetection algorithm in this chapter, four groups of ablation experiments were set up on the LaSOT test dataset for analysis and comparison. The four groups of experiments are basic tracking algorithm SiamRPN, long-term target tracker ASiam grounded in redetection and basic tracking algorithm, long-term target tracker TSiam based on adaptive template update and basic tracking algorithm, and long-term target tracker LTUSiam based on redetection, adaptive template update and basic tracking algorithm.

The experimental results are presented in FIG. 12. LaSOT test dataset uses success rate and accuracy rate to assess the tracking effectiveness of the algorithm. The figure demonstrates that, in comparison to the benchmark tracking algorithm

SiamRPN, the success rate and accuracy of the ASiam tracker with the redetection module increased by 0.02 and 0.012. The success rate and accuracy of TSiam tracker with the addition of adaptive template update increased by 0.014 and 0.01. In comparison to the benchmark algorithm, the success rate and accuracy of the algorithm based on template update and redetection are improved by 0.032 and 0.022. Experiments show that LTUSiam, the long-term target algorithm presented in this chapter, tends to take the lead in the success rate and accuracy of long-term sequences, and effectively improves the tracking performance.



(a)Success rate graph



(b) Accuracy graph

Figure 12. Ablation experiment results

IV. Conclusions

The LTUSiam algorithm, based on SiamRPN, integrates an adaptive template update module and a redetection module. It employs a three-level cascade gated cycle unit to extract timing information, including geometric, discriminative, and appearance features, while using local anomaly information to assess the target state and update the template to prevent sample contamination.

For global search, the algorithm utilizes a template matching-based redetection method to quickly and accurately locate lost targets. An evaluation score sequence combines the initial target template with a confidence score to determine if a target has been lost and to switch tracking states as needed. Experiments on the VOT2018_LT dataset show that LTUSiam operates at 28 frames per second and achieves an F-value of 0.644, demonstrating effective long-term tracking performance, particularly in occlusion and out-of-view scenarios.

While LTUSiam dynamically updates its template to adapt to target appearance changes, enhancing local tracking accuracy, performance may decline under extreme lighting changes or complex backgrounds. Although the adaptive update and redetection modules improve occlusion handling, their effectiveness can be limited during prolonged severe occlusion or complete target disappearance. Future developments could include utilizing deeper convolutional neural networks (CNNs) for feature extraction to better handle complex backgrounds and lighting variations, integrating visual data with other sensors (such as depth sensors or infrared sensors) to enhance stability, and exploring methods to maintain efficient tracking across different scenes and conditions.

References

[1] S. M. Marvasti-Zadeh, L. Cheng, H. Ghanei-Yakhdan, and S. Kasaei, “Deep Learning for Visual Tracking: A Comprehensive Survey,” IEEE Transactions on Intelligent Transportation Systems, vol. 23, no. 5, pp. 3943–3968, 2022, doi: 10.1109/TITS.2020.3046478.

[2] Y. Zhang, T. Wang, K. Liu, B. Zhang, and L. Chen, “Recent advances of single-object tracking methods: A

brief survey," Neurocomputing, vol. 455, pp. 1–11, 2021, doi:https://doi.org/10.1016/j.neucom.2021.05.011.

[3] J. Zhang, J. Sun, J. Wang, and X.-G. Yue, "Visual object tracking based on residual network and cascaded correlation filters," Journal of Ambient Intelligence and Humanized Computing, vol. 12, no. 8, pp. 8427–8440, Aug. 2021, doi: 10.1007/s12652-020-02572-0.

[4] F. Chen, X. Wang, Y. Zhao, S. Lv, and X. Niu, "Visual object tracking: A survey," Computer Vision and Image Understanding, vol. 222, p. 103508, 2022, doi: https://doi.org/10.1016/j.cviu.2022.103508.

[5] J. Chai, H. Zeng, A. Li, and E. W. T. Ngai, "Deep learning in computer vision: A critical review of emerging techniques and application scenarios," Machine Learning with Applications, vol. 6, p. 100134, 2021, doi: https://doi.org/10.1016/j.mlwa.2021.100134.

[6] K. Tong and Y. Wu, "Deep learning-based detection from the perspective of small or tiny objects: A survey," Image and Vision Computing, vol. 123, p. 104471, 2022, doi: https://doi.org/10.1016/j.imavis.2022.104471.

[7] Y. Zhang, L. Wang, D. Wang, J. Qi, and H. Lu, "Learning Regression and Verification Networks for Robust Long-term Tracking," International Journal of Computer Vision, vol. 129, no. 9, pp. 2536–2547, Sep. 2021, doi: 10.1007/s11263-021-01487-3.

[8] E. Tian, Y. Lei, J. Sun, K. Zhou, B. Zhou, and H. Li, "The Segmentation Tracker With Mask-Guided Background Suppression Strategy," IEEE Access, vol. 12, pp. 124032–124044, 2024, doi: 10.1109/ACCESS.2024.3451229.

[9] K. Dai, Y. Zhang, D. Wang, J. Li, H. Lu, and X. Yang, "High-Performance Long-Term Tracking With Meta-Updater," in Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), Jun. 2020.

[10] L. Huang, X. Zhao, and K. Huang, "GlobalTrack: A Simple and Strong Baseline for Long-Term Tracking," AAAI, vol. 34, no. 07, pp. 11037–11044, Apr. 2020, doi: 10.1609/aaai.v34i07.6758.

[11] H. Fan et al., "LaSOT: A High-quality Large-scale Single Object Tracking Benchmark," International Journal of Computer Vision, vol. 129, no. 2, pp. 439–461, Feb. 2021, doi: 10.1007/s11263-020-01387-y.

[12] R. Faster, "Towards real-time object detection with region proposal networks," Advances in neural information processing systems, vol. 9199, no. 10.5555, pp. 2969239–2969250, 2015.

[13] L. Bertinetto, J. Valmadre, J. F. Henriques, A. Vedaldi, and P. H. S. Torr, "Fully-Convolutional Siamese Networks for Object Tracking," in Computer Vision – ECCV 2016 Workshops, G. Hua and H. Jégou, Eds., Cham: Springer International Publishing, 2016, pp. 850–865.

[14] B. Li et al., "Evolution of siamese visual tracking with very deep networks," in Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Long Beach, CA, USA, 2019, pp. 15–20.

[15] M. Zolfaghari, K. Singh, and T. Brox, "Eco: Efficient convolutional network for online video understanding," in Proceedings of the European conference on computer vision (ECCV), 2018, pp. 695–712.

[16] A. Lukežič, L. Č. Zajc, T. Vojíř, J. Matas, and M. Kristan, "Now you see me: evaluating performance in long-term visual tracking," arXiv preprint arXiv:1804.07056, 2018.

[17] L. Leal-Taixé, A. Milan, I. Reid, S. Roth, K. M. Schindler, and M. Challenge, "Towards a benchmark for multi-target tracking," arXiv preprint arXiv:1504.01942, vol. 34, 2015.

A Baseline for Violence Behavior Detection in Complex Surveillance Scenarios

Yingying Long
School of Computer Science and Engineering
Xi'an Technological University
Xi'an, China
E-mail: longyingying@st.xatu.edu.cn

Zongxin Wang
School of Computer Science and Engineering
Xi'an Technological University
Xi'an, China
E-mail: zongxinwang@yeah.net

Hanzhu Wei
School of Computer Science and Engineering
Xi'an Technological University
Xi'an, China
E-mail: weihanzhu@st.xatu.edu.cn

Xiaojun Bai
School of Computer Science and Engineering
Xi'an Technological University
Xi'an, China
E-mail: baixiaojun@st.xatu.edu.cn

***Abstract*—Violence detection can improve the ability to deal with emergencies, but there is still no data set specifically for violence detection. In this work, we propose VioData, a datasets specialized for detection in complex surveillance scenarios, and to more accurately assess the efficacy of these datasets, we propose a violence detection model based on target detection and 3D convolution. The model consists of two key modules: spatio-temporal feature extraction module and spatio-temporal feature fusion module. Among them, the spatio-temporal feature extraction module consists of a spatial feature module that extracts key frames using ordinary convolutional networks and a temporal feature extraction module that establishes temporal features using 3D convolution. The spatio-temporal feature fusion module Channel Fusion and Attention Mechanism (CFAM) fuses the temporal and spatial features. The experimental results indicate that the precision of the suggested detection model on UCF101-24, JHMDB behavioral detection datasets, and our proposed violence detection datasets, VioData, is improved compared to other violence detection models, which not only verifies the validity of the datasets, but also provides a baseline for the subsequent research and improvement in this area.**

Keywords-Violent Behavior Detection; Datasets; Spatio-temporal Feature; Target Detection; Feature Fusion

I. INTRODUCTION

Violent behavior is defined as the use of force and other means to harm oneself or others, and violent behavior detection can serve as one of the roles to meet the growing public safety needs. Utilizing deep learning technologies in the domain of violent behavior detection can capture eligible violent behaviors from cameras and alert the police, which is a useful tool for public security officers' daily tasks.

However, violent behaviors mostly occur outdoors, and in complex surveillance scenes with large field of view outdoors, the small size of the human target makes it challenging to locate the important parts of the body, many occlusions, and the complex background, which poses a great challenge to the detection of violent behaviors. In the existing public behavior detection datasets UCF101-24 and JHMDB, which contain 45 categories of more common behaviors, there is no violence detection datasets specifically for complex surveillance scenes. Moreover, most of the existing behavior detection algorithms use a two-stage strategy, such as SlowFast [9] and other candidate areas are initially generated by two-stage detection algorithms, and then finally perform feature extraction and classification on the

candidate regions to ultimately determine the behavioral categories and locations. However, two-stage algorithms have been difficult to apply in complex surveillance scenarios, firstly, the method of obtaining candidate frame sequences through the detection algorithm cuts off the potential relationship between people and people, people and background, etc. Finally, the operation of analyzing all detected people is challenging to fulfill the real-time requirements in reality.

Therefore, this paper collects publicly available surveillance videos of public places and takes them as the research object, and uses them as the raw data to produce a set of violence detection datasets, VioData, which is specialized in complex surveillance scenes; and offers a violence detection module utilizing target identification and three-dimensional convolutional networks. and target detection for accomplishing the violence detection task more efficiently. The module integrates the spatio-temporal feature data of the video sequence and extracts the spatial properties of the key frames through ordinary convolutional network, extracts temporal characteristics from the video using a 3D convolutional network, and finally fuses the spatio-temporal features through spatio-temporal feature fusion network. In addition, the module UCF101-24, the JHMDB datasets, and the VioData datasets constructed in this paper on which extensive experiments were carried out, and the experimental findings verify the effectiveness of the datasets and the module's ability to produce competitive outcomes in the detection of violent behavior in complex outdoor scenes. The main contributions of this paper are as follows:

- VioData, a datasets specialized for violence detection.
- Because of the occlusion phenomenon in complex violent behavior scenes, a temporal feature extraction network is proposed in this paper. which introduces 3D Convolutional Block Attention model (3D-CBAM) attention mechanism and spatio-temporal depth separable convolution to better utilize the information between consecutive frames to better extract the features in the video sequences, and to improve how the network perceives the foreground features; secondly, to detect the aggressive behavior more precisely, the introduced Atrous Spatial Pyramid Pooling (ASPP) model is introduced in order to more accurately detect violent acts, and the fusion of feature maps of different sensory fields is obtained by utilizing different scales of convolution.
- In order to naturally fuse spatio-temporal information for a later, more precise identification of aggressive behavior, a spatio-temporal feature fusion module was designed.

II. Related Work Research

We will review the work related to behavioral detection datasets and review the work on techniques used for behavioral detection from four perspectives: behavioral detection based on traditional features, behavioral detection based on recurrent neural networks, behavioral detection based on multi-stream neural networks, and behavioral detection based on three-dimensional convolutional networks.

A. Behavioral detection datasets

Behavior detection datasets typically contain data collected from sources such as videos, sensors, etc. And are used to train and test algorithms for recognizing and analyzing human behavior. The UCF101 [1] datasets is among the biggest datasets of human behavior that are currently accessible, containing 101 action categories, almost 13,000 video snippets, totaling 27 hours of footage. Real user-uploaded films with crowded backdrops and camera motions make up the database. HMDB with Joint Annotation (JHMDB) [2] datasets A subset of the Human Metabolome Database (HMDB) [3] datasets contains 21 action categories, each involving the movement of a single character. The dataset was annotated with 2D joint model, providing information on the character's pose, optical flow, and segmentation for analyzing action recognition algorithms. The Kinetics [4] datasets is a human action video datasets introduced by DeepMind that contains 400 human action categories, each with 400 video snippets, each lasting roughly 10 seconds, from different YouTube videos. The dataset covers a wide range

of action categories, including human-object interaction and human-human interaction.

B. Behavior detection based on traditional features

Before the popularization of deep learning techniques, researchers used traditional features to process image information. The technique mostly included manually removing characteristics from video frames, which were then fed into support vector machines and decision trees for further behavioral analysis and identification. Xu [5] et al. suggested a technique for detecting violent videos that uses sparse coding and MoSIFT characteristics. Initially, the low-level description of the video is extracted using the MoSIFT algorithm, then feature selection is performed by Kernel Density Estimation (KDE) to eliminate noise, and finally the selected MoSIFT features are further processed using a sparse coding scheme to obtain highly discriminative video features. Febin [6] proposed a new descriptor Motion Boundary SIFT (MoBSIFT) to more effectively identify the characteristics of violent actions in the video. This module is able to filter out the random motions in the nonviolent behaviors, and represent and classify the violent videos by sparse coding technique, which has high accuracy and robustness in detecting violent behaviors.

C. Recurrent neural network based behavior detection

By receiving the hidden state of the preceding moment, a recurrent neural network (RNN) may model the frames in a movie as an ordered sequence, which affects the state of the next moment, and the extracted temporal features are able to express human behavior. With networks like Long Short-Term Memory (LSTM), this behavior detection technique first extracts spatial data from the ordered sequence of frames, and then it goes on to extract temporal features from the video. Sudhakaran [7] proposed ConvLSTM, which aggregates frame-level violent behavioral features in the video by capturing the spatio-temporal features and captures the differences between consecutive frames by computing the motion changes, which reduces the amount of data to be processed. Liang [8] et al. used GhostNet and ConvLSTM to construct a long-term recurrent convolutional network and introduced a multiple attention mechanism in the video preprocessing stage to enhance the attention to the key information in the video, which improves the ability of detecting violent behavior in the video.

D. Behavior detection based on multi-stream neural networks

Multi-stream neural networks usually have many branches, before employing a classifier to identify behaviors, each branch independently extracts many feature streams from a large number of samples and aggregates the extracted features. Feichtenhofer [9] et al. designed a SlowFast network based on frame rate speed. The network contains two paths, Slow path and Fast path, to extract spatial semantic information and motion information at lower and higher frame rates, respectively, to enhance behavior detection. Next, Okan [10] proposed a multi-modal parallel module You Only Watch Once (YOWO) based on a dual channel structure. The network has two branches: one uses 2D-CNN to extract the spatial properties of key frames, while the other uses 3D-CNN to extract the spatio-temporal features of the video segment made up of earlier frames, and finally, fuses the features using channel fusion and the attention mechanism to perform frame-level detection for behavioral Localization of actions. Li [11] et al. suggested a novel technique for detecting violence based on a multi-stream detection model, which combines three distinct streams—a temporal stream, a local spatial stream, and an attention-based spatial RGB stream—to improve the performance of violent behavior recognition in videos. Islam [12] et al. suggested an effective dual-stream deep learning architecture using pre-trained MobileNet and LSTM (SepConvLSTM), in which one stream manages frame background suppression and the other handles frame differences between neighbors. In order to provide discriminative features that aid in differentiating between violent and nonviolent activities, a straightforward input preprocessing technique highlights moving objects in the frames while suppressing the nonmoving background and recording the inter-frame actions.

E. Behavior detection based on 3D convolutional networks

Conventional 2D convolutional neural networks that have been trained on single-frame images are unable to reflect the correlation between consecutive frames, while 3D convolutional networks are able to directly extract frames from the video, and then fed into 3D CNNs to extract the spatio-temporal features in the frame sequences, the network learns the characterization of the behaviors in the video after multilayered convolutional and pooling operations, and accurately detects the behaviors in the video, and it is currently an important research direction. Carreira [13] based on the Inception network and extended it from 2D to 3D, proposed the network Inflated 3D ConvNet (I3D) which is able to process video data for behavioral detection. Direkoglu [14] computes optical flow vectors for every frame to produce a motion quantum image (MII), It is then used to train a Convolutional Neural Network (CNN) to identify abnormal behavioral events in a crowd. The proposed MII is mainly based on the optical flow magnitude and angular difference calculated from the optical flow vectors in consecutive frames, which helps to distinguish between normal and abnormal behavior. Dong [15] et al. suggested the attentional residual 3D network (AR3D) and the residual 3D network (R3D), which were model ed by upgrading the current 3D CNNs by adding the residual structure and attention mechanism, The behavior detection performance of the model has been improved in different degrees. Li [16] et al. establish a 3D-DenseNet dense connectivity model , extract spatio-temporal features using 3D-DenseNet algorithm, redistribute the weights of each feature using the Squeeze-and-Excitation Networks (SENet) channel attention model , and then use the transition layer sampling, and then pass the outcomes to the fully connected layer using the global average pooling technique to finish the violence detection task. XU [17] et al. proposed the SR3D algorithm, which adds a BN layer before the 3D convolutional operation and presents the ReLu activation mechanism to enhance the network's learning capabilities while, extends the SE attention mechanism to 3D by introducing it into the 3D convolutional model and boosts the weights of the important channels, which improves the ability to detect the human behaviors in the video in the network.

III. VIOLENCE TEST DATASETS PRODUCTION

Because there are no samples of datasets dedicated to violence detection in the current public datasets in the field of video behavior detection such as UCF101-24, JHMDB and Kinectics. Therefore, in this paper, we produce VioData, a violence detection datasets specialized for complex surveillance scenarios.

First, this paper collects about 1500 video clips of violent behavior from publicly available real surveillance video data.

Second, since the length of the collected surveillance videos varies between 1-10 minutes and there are not many clips in which violent behaviors occur, the collected surveillance videos are manually cropped to segment the videos into short videos of violent behaviors of about 10 seconds. Then, the obtained short videos were subjected to frame extraction, before extracting the frames, the videos were converted into easily labeled RGB image sequences and the blurred images were discarded, and the extracted video frames were deposited into a separate folder to obtain a separate clip of violent behavior using a frequency of 1 frame every 5 frames.

Finally, the human targets of violent acts in the video are labeled with frame-level truth frames using the LabelImg tool, based on the collected violent actclip clips, the manual labeling method is used, the violent act targets are labeled with rectangular frames, and the targets with more than 50% occlusion are not labeled, and the violent act targets of the part-frame Pictures are labeled as Fig.1 illustrates. The labeled information is saved as an XML file, and the xml file contains the image file address, the truth frame coordinate information and the behavioral category of the target.

(a)abuse

(b)attack

(c)fight

Figure 1. Illustration of a sample of labeled acts of violence

IV. Methodology Of This Paper

The framework of the violence detection module is shown in Fig.2, the framework has two branches of inputs, the output is a series of video frames with a violence detection frame containing the outcomes of the violence category, while the first branch consists of a series of video frames and the second branch consists of extracted keyframes. There are three modules in the module: one for spatiotemporal feature extraction, one for spatiotemporal feature fusion, and the structure of the spatio-temporal feature extraction model consists of an I3D network and a CSPDarkNet-Tiny network for extracting spatial features. The 3D convolution-based I3D network video is used for temporal modeling and for extracting temporal features; the CSPDarkNet-Tiny network model is the 2D features of the keyframes and is used for extracting the spatial features of the keyframes. The temporal and spatial feature fusion model integrates the feature information of the two branches and filters the valid information among them, lastly, to obtain the violence detection findings, the fused feature map results are input into the prediction head output.

A. Spatio-temporal feature extraction module

1) Timing feature extraction module

Violence detection for complex surveillance scenarios requires high real-time modeling, and occlusion phenomena are likely to occur in the violence scenarios. The 3D Inception (3D) Inc model in the Inflated 3D ConvNet (I3D) network uses ordinary 3D convolution, but its computational overload makes it difficult to perform real-time violence detection. The original Inflated 3D ConvNet (I3D) network is prone to omission and false detection when detecting violence with occlusion phenomenon. Therefore, in this work, according to the features of the original I3D network structure, spatio-temporal depth-separable convolution and 3D-CBAM attention are introduced to improve both efficiency and accuracy.

In terms of real-time, after frame-by-frame convolution operation, the spatio-temporal information is combined by point-by-point convolution to extract higher-level feature representation in real time. The improved 3D Inc reduces the computational effort of the 3D Inc module exponentially by replacing the standard $3 \times 3 \times 3$ convolution in the middle two branches with spatio-temporal depth-separable convolutions of $1 \times 3 \times 3$ and $3 \times 1 \times 1$ shapes. The 3D Inc module finally fuses the features of the four branches. The structural diagram of the optimized 3D Inc network is shown in (c) in Fig.2.

In terms of accuracy, the Convolutional Block Attention model (CBAM)[18], which aggregates the temporal dimension information based on CBAM, is introduced in this study since the temporal information in the video sequences cannot be properly utilized. The structure diagram of 3D-CBAM is shown in (b) in Fig.2. The Channel Attention model (CAM) and the Spatial Attention model (SAM) make up 3D-CBAM. The Channel Attention model processes the input feature map F3D to produce the channel weight vector. which is multiplied with F3D to obtain the

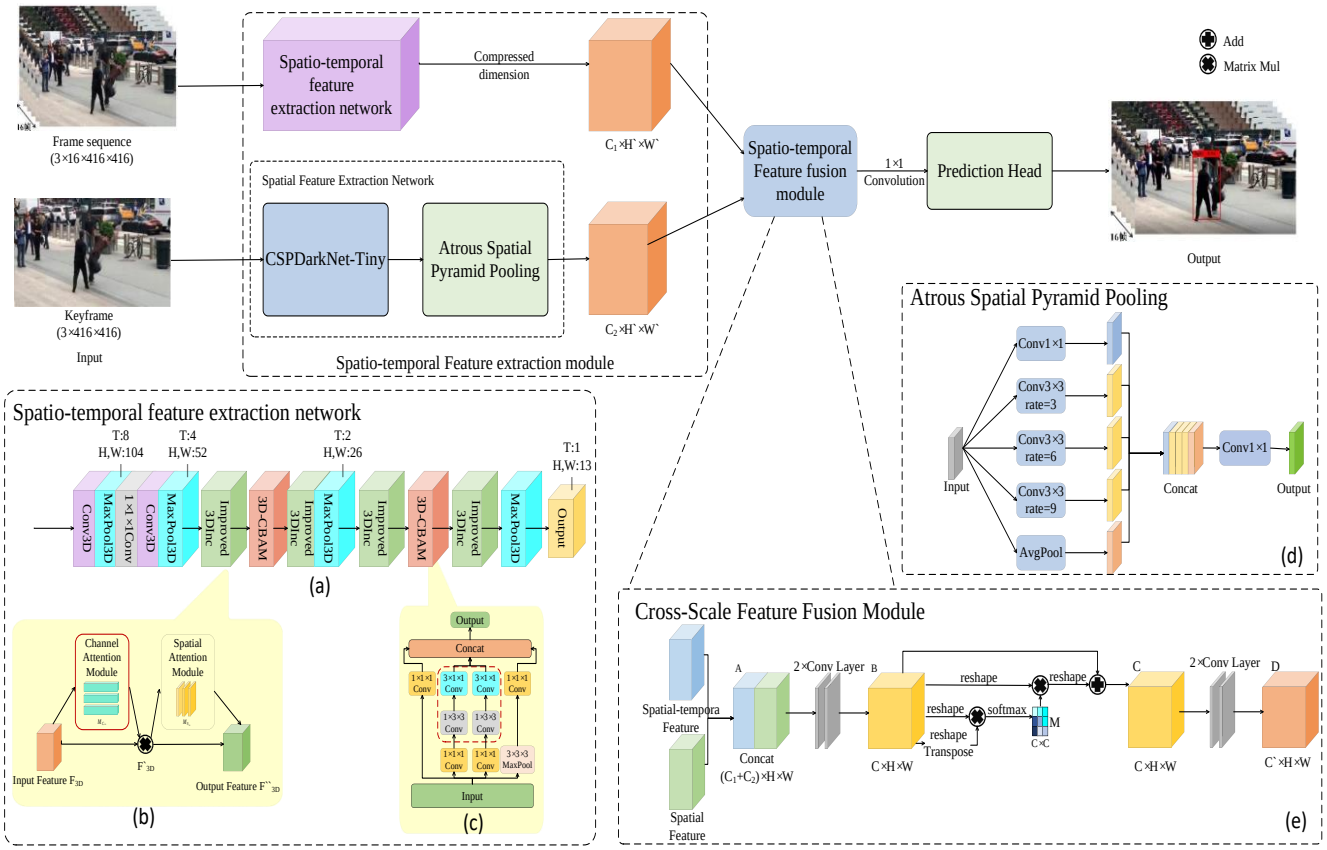


Figure 2. The violence detection algorithm's framework is displayed in Fig.2. The model for extracting spatio-temporal features and the spatio-temporal feature fusion module make up the majority of the framework. The spatio-temporal feature extraction model is composed of the temporal feature extraction model and the spatial feature extraction module, and the I3D network is the network structure of the temporal feature extraction model, as illustrated in (a); (b)(c) are the 3D-CBAM Attention Mechanism and 3D Inception (3D Inc) module, respectively. The Atrous Spatial Pyramid Pooling (ASPP) model is added at the end of the spatial feature extraction model, which has the CSPDarkNet-Tiny network as its network structure, which is shown in (d), where rate denotes the expansion rate of the null convolution. atrous Spatial Pyramid Pooling (ASPP) has five branches, including one ordinary convolutional branch, three null convolutional branches, and one global average pooling branch; (e) shows the overall structure of Channel Fusion and Attention Mechanism(CFAM); D is the final output feature map of CFAM, and C1 and C2 are the number of feature map output channels for the I3D network and the ASP module, respectively.

F'_{3D} weighted feature map. The Spatial Attention model then processes F'_{3D} to get the spatial weight, which is then multiplied by the feature F'_{3D} to get the final feature F''_{3D}, which combines spatial and channel attention. channel and spatial focus of the F''_{3D} feature. The following is the computational expression for 3D-CBAM:

$$F'_{3D} = M_{C_{3D}}\left(F_{3D}\right) \otimes F_{3D} \tag{1}$$

$$F''_{3D} = M_{S_{3D}}\left(F'_{3D}\right) \otimes F'_{3D} \tag{2}$$

where $M_{S_{3D}}$ represents the spatial attention, and $M_{C_{3D}} \in R^{C \times D \times 1 \times 1}$ represents the channel attention. D is the number of frames in the video sequence frame, while C is the number of feature map channels. In Fig.2, the enhanced I3D network structure is displayed in (a).

2) Spatial feature extraction module

Wang [19] et al proposed CSPDarkNet combines the features of Cross Stage Partial Network (CSP) structure and DarkNet framework, which is able to maintain or even improve the capability of CNN while reducing the amount of computation. In this paper, considering the scenario of violence detection, we need to use an efficient and lightweight network, so we chose a lightweight CSPDarkNet network, CSPDarkNet-Tiny, its network structure is shown in Fig.3, but because of the violence detection method, the lightweight network may lead to insufficient computational power to deal with occlusion or

background complexity, which leads to the decrease of accuracy. Therefore, this paper introduces CSPDarkNet-Tiny's last layer is supplemented with the Atrous Spatial Pyramid Pooling (ASPP) module. The ASPP input feature maps are branched through five null convolutions to obtain feature maps with five different sensory fields, which are spliced and fused along the channel dimensions, and then adjusted using a 1× 1 convolutional number of channels to acquire more specific visual information. Fig.2(d) displays the ASPP model's structure.

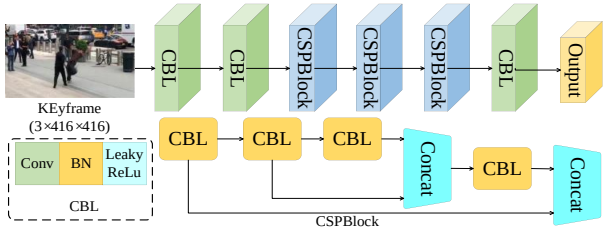


Figure 3. CSPDarkNet-Tiny Network Overall Structure

B. Spatio-temporal feature fusion module

The temporal fusion attention model(TFAM) [19] is an attention mechanism module for improved video object detection, which improves object representation by combining multi-frame and single-frame attention modules and dual-frame fusion modules, but it has too much computation and weak generalization ability, which is not conducive to the application of violent behavior detection. Therefore, Channel Fusion and Attention Mechanism (CFAM) is introduced in this paper to effectively integrate the temporal features obtained by I3D network with the spatial features obtained by CSPDarkNet-Tiny network to record the inter-channel dependencies. Fig.2(e) displays the CFAM model's structure diagram.

The following is how feature fusion specifically works: firstly, the feature maps obtained from the first two networks are spliced to obtain the feature map $A \in R^{(C1+C2)\times H\times W}$, then the correlation between the feature maps is captured using the local receptive fields of the convolutional layers, and the correlation feature map $B \in R^{C\times H\times W}$ is produced by passing the feature map A through two convolutional layers. Since direct correlation calculation would make the computation complicated, a reshaping operation is performed on B to obtain a reshaped feature map F. The elements of each channel in the feature map are converted into one-dimensional vectors to simplify the computation. The expression is as follows:

$$B \in R^{C\times H\times W} \xrightarrow{\text{vectorization}} F \in R^{C\times H\times W} \tag{3}$$

First, the resulting feature map F is dot-producted with its transposed feature map F^T to obtain a covariance matrix $G \in R^{C\times N}$, where $N=H\times W$. This matrix reveals the correlation between different features. Its expression is as follows:

$$G = F \times F^T \tag{4}$$

$$G_{i,j} = \sum_{k=1}^{N} F_{ik} \times F_{jk} \tag{5}$$

Where $G_{i,j}$ represents the inner product between the feature map F and F^T. After that, the resulting matrix G is subjected to *softmax* operation to generate the channel attention feature map $M \in R^{C\times C}$. The *softmax* function is able to transform the values between the range of 0-1, which represents the attention weight of each position. the expression of M feature map is as follows:

$$M_{i,j} = \frac{e^{G_{ij}}}{\sum_{j=1}^{C} e^{G_{ij}}} \tag{6}$$

In order for the attention map M to have an effect on the original feature map, the matrix F' is obtained by dot-product multiplication of M with the reshaping matrix F, which makes the features of the parts with high weights more prominent. Then F' is reshaped to $F'' \in R^{C\times H\times W}$ of the same size as B.

$$F' = M \times F \tag{7}$$

To alleviate the gradient vanishing problem and accelerate the model convergence, F'' is multiplied with the hyperparameter α and superimposed with the feature map B using the expression in (8) to get the feature map $C \in R^{C\times H\times W}$, the final spatio-temporal feature map $D \in R^{C\times H\times W}$ with the

attention weights is obtained by consecutively applying two convolutions to the resultant feature map *C*.

$$C = \alpha \times F'' + B \tag{8}$$

C. Loss function

The loss function proposed in this paper contains three components: classification prediction loss L_{cls}, localization loss L_{rect}, and confidence loss L_{obj}.

The classification prediction loss formula is as follows:

$$\mathrm{y_i} = \mathrm{sigmoid}\left(\mathrm{x_i}\right) = \frac{\mathrm{e}^{\mathrm{x_i}}}{\sum_{\mathrm{n=1}}^{\mathrm{N}} \mathrm{e}^{\mathrm{x_n}}} \tag{9}$$

$$L_{cls}\left(y, y_i\right) = -\frac{1}{N}\sum_{n=1}^{N} L_{BCE_{cls}} \tag{10}$$

$$L_{BCE_{cls}} = y \times \log\left(y_i\right) + (1-y) \times \log\left(1-y_i\right) \tag{11}$$

where x_{i} is the category's projected value and *N* is the total number of categories in the datasets, y_i is the current category probability, and *y* is the true value of the current category.

The localization loss formula is as follows:

$$v = \frac{4}{\pi}\left(\arctan\frac{w^{gt}}{h^{gt}} - \arctan\frac{w}{h}\right)^2 \tag{12}$$

$$\alpha = \frac{v}{1 - \mathrm{IoU}\left(B, B_{gt}\right) + v} \tag{13}$$

$$L_{rect}\left(B, B_{gt}\right) = CIoU\left(B, B_{gt}\right) - \frac{p^2(B, Bgt)}{c^2} - \alpha \tag{14}$$

Where w^{gt} as well as h^{gt} are the target real frame's width and height, the target predicted frame's width and height, *v*, which is the value of the projected frame normalized by extrapolating the width-to-height ratio of the predicted frame to the actual frame, and p^2, which is the distance between the predicted frame's centroid and the real frame's centroid, where α represents the balance between the loss resulting from the measurement of aspect ratio and the loss due to IoU. The confidence loss is publicized as follows:

$$L_{obj}\left(C, C_i\right) = -\frac{1}{N}\sum_{n=1}^{N} L_{BCE_{obj}} \tag{15}$$

$$L_{BCE_{obj}} = C \times \log\left(C_i\right) + (1-C) \times \log\left(1-C_i\right) \tag{16}$$

where *C* is the current grid region's confidence, C_i is the expected value of confidence, and *N* is the number of feature points.

The three loss functions above are integrated to get the total loss function with the following formula:

$$L = \alpha_1 \times L_{cls} + \alpha_2 \times L_{rect} + \alpha_3 \times L_{obj} \tag{17}$$

To ensure that the weights of the various loss terms are balanced, the hyperparameters α_1, α_2 and α_3 are set. where α_1 has a value of 0.4, α_2 has a value of 0.3 and α_3 has a value of 0.3.

V. Experiments And Analysis Of Results

A. Experimental setup

The Kinectics datasets are used to train the model suggested in this research, and the custom datasets VioData are used to refine it.

In order to be able to enrich the training set and make the model better acquire the effective features in the video frames, three data enhancement operations are adopted in this paper, including horizontal flipping, random scaling, and color enhancement. The data enhancement operations expand the datasets, reduce overfitting, enhance the generalization ability of the model, and improve the robustness of the model.

The training settings are displayed in Table Ⅰ below.

TABLE I. Parameter settings in network training

Parameter	Setting
Initial Learning Rate	0.001
Epoch	230

Parameter	Setting
ReSize	（416,416）
Weight Decay	0.0005
Optimizer	Adam

Configure the model parameters to be saved once per ten iterations while the model is being trained, and save the output of the training loss and validation loss once at the completion of each epoch. When the loss iteration reaches 180 rounds, the training loss is still decreasing, but the loss of the validation set starts to rise, indicating that the model has been overfitted, so the model parameters at the end of the 180th epoch are saved as the optimal parameters.

The effect of the network model for violence detection on the VioData datasets is visualized as shown in Fig.3, where a video of a violent act is subjected to model inference to obtain the location of the violent act and its category, proving the effectiveness of the VioData datasets.

Figure 4. Violence detection results

B. Experimental results and analysis

To compare with other violence detection techniques and show the efficacy of the suggested enhanced modules in the suggested violence detection model, we conducted numerous tests in this work. The experiments are conducted on three datasets (UCF101-24, JHMDB, and VioData).

1) Experimental result and analysis

In order to confirm the model's efficacy for violence detection, the model put out in this work is contrasted with current behavioral detection techniques in this section. The following four models are chosen for comparison studies:

a) MPS [21]: this model proposes a new fusion strategy that not only fuses the appearance and optical flow information of dual-stream networks, but also includes a solution to the problem of small camera movements.

b) P3D-CTN [22]*:* the core idea of this model is to use the so-called Pseudo-3D Convolution, which is a method that combines 2D spatial convolution with 1D temporal convolution. This method can effectively extract spatio-temporal features from videos without significantly increasing the computational complexity.

c) STEP [23]*: t*his model contains two main parts, spatial refinement and temporal expansion. Each step in spatial refinement uses the regression output of the previous step to improve the quality; temporal extension focuses on improving the accuracy of action classification through the duration of the video clip.

d) YOWO [10]*: t*his architecture contains two branches, one for extracting spatial features of key frames and the other for modeling the spatio-temporal features of video clips consisting of previous frames, and finally the features obtained from the two branches are fused through the attentional mechanism and regressed for classification.

The outcomes of this comparison experiment are displayed in the Table Ⅱ:

TABLE II. RESULTS OF VIOLENCE DETECTION ACCURACY OF DIFFERENT MODELS

Method	UCF101-24	JHMDB	VioData
	mAP		
MPS	82.4%	-	85.3
P3D-CTN	-	84.0%	84.9%
STEP	83.1%	-	86.4%
YOWO	82.5%	85.7%	88.0%
ours	89.8%	88.6%	91.8%

2) Ablation experiments

In this part, we use a series of ablation experiments to assess how various network enhancements affect the effectiveness of video behavior identification.

First, we introduce the ASPP model on the CSPDarkNet-Tiny backbone network, and next, we introduce spatio-temporal depth-separable convolution in the I3D network, and further

experiments are conducted on the same datasets. The experimental results are shown in Table Ⅲ.

TABLE III. DETECTION RESULTS WITH EMBEDDED ASPP MODEL AND INTRODUCTION OF SPATIO-TEMPORAL DEPTH SEPARABLE CONVOLUTION

Network	UCF101-24	JHMDB	VioData
	mAP		
Baseline	78.5%	75.3%	78.9%
CSPDarkNet-Tiny+ASPP	80.7%	76.6%	82.0%
CSPDarkNet-Tiny+ASPP++I3D(Improved 3D Inc)	84.8%	80.4%	86.5%

Table Ⅲ makes it clear that the ASPP paradigm was introduced in the CSPDarkNet-Tiny network has an improvement of 2.2, 1.3, and 3.1 percentage points on the three datasets, respectively, which indicates that the ASPP model is effective in improving the detection accuracy of the model . By introducing spatio-temporal depth-separable convolution to improve the I3D network, the model accuracy has an improvement of about 4 percentage points on all three datasets, indicating the effectiveness of spatio-temporal depth-separable convolution in improving the detection accuracy.

Finally, we embedded the 3D-CBAM attention model in the improved I3D network and conducted experiments at different embedding locations. Table Ⅳ displays the findings of the experiment.

TABLE IV. DETECTION RESULTS OF 3D-CBAM ATTENTION MODEL EMBEDDED AT DIFFERENT LOCATIONS

Network	Embedding position	UCF101-24	JHMDB	VioData
		mAP		
I3D	-	84.4%	80.4%	86.5%
	3D Inc_1	86.1%	83.7%	89.0%
	3D Inc_2	86.7%	83.3%	88.3%
	3D Inc_3	85.9%	84.2%	89.6%
	3D Inc_1+3D Inc_2	88.2%	87.5%	90.7%
	3D Inc_1+3D Inc_3	89.8%	88.6%	91.8%
	3D Inc_2+3D Inc_3	88.0%	88.0%	91.4%
	3D Inc_1+3D Inc_2+3D Inc_3	90.0%	88.7%	92.0%

As seen in Table 3.3, the addition of the 3D-CBAM attention model has a corresponding improvement on all three datasets, and embedding more than one will give a further improvement over embedding one. Among them, adding the attention model after the first, second and third 3D Inc modules performs the best on all three datasets, but due to the consideration of the amount of parameter computation, adding the 3D-CBAM attention after the first and third 3DInc not only gives better accuracy, but also keeps the network's computation from being overly large to satisfy the requirements of video detection.

VI. CONCLUSIONS

Aiming at the problem that there is no specific violence detection data set in complex surveillance scenarios, this paper collects 1,500 violence surveillance videos in public data sets, filters and extracts the collected videos, and manually marks each frame to obtain violence detection data set VioData. This work suggests a violence detection module based on target identification and 3D convolution to deal with opacity and ambiguous human targets while detecting violence in intricate surveillance situations. This work suggests a violence detection module based on target detection and 3D convolution for detection in complex surveillance scenarios with occlusion issues and ambiguous human targets. To enhance the capacity to extract human traits from key frames, the ASPP module is incorporated into the network architecture; the 3D Inc module is improved to minimize the amount of network parameters; and by embedding the 3D-CBAM attention mechanism, the network is able to focus more on detecting the key regions of violent behavior based on the weight of the feature map. In the experimental phase, this paper first verifies whether the ASPP model is effective, followed by a comparative analysis of the 3D Inc model before and after optimization. Prior to model training, data augmentation operations are carried out on the video data to increase the model's capacity for generalization. The experimental results demonstrate that the approach suggested in this paper can successfully improve the precision of violence detection, verify the validity of the

datasets and propose benchmarks for researchers to improve the enhancement.

Considering that the experimental data is still limited, the scenes in the video data are not rich and complex enough, and the crowd violence category is not rich enough. In the future, we will continue to collect videos and look for datasets with more complex and diverse backgrounds that contain multiple violence categories.

ACKNOWLEDGMENT

This work was supported by the College Students' Innovative Entrepreneurial Training Plan Program (No. S202310702107).

REFERENCES

[1] Soomro K, Zamir A R, Shah M. UCF101: A Datasets of 101 Human Actions Classes from Videos in The Wild [J]. Computer Science, 2012.DOI: 10.48550/arXiv.1212.0402.

[2] Jhuang H, Gall J, Zuffi S, et al. Towards understanding action recognition [C] //IEEE International Conference on Computer Vision. IEEE, 2014. DOI: 10.1109/ICCV.2013.396.

[3] Wishart D S, Djoumbou F Y, Ana M, et al. HMDB 4.0: the human metabolome database for 2018 [J]. Nucleic Acids Research, 2017(D1): D1.DOI: 10.1093/nar/gkx1089.

[4] Kay W, Carreira J, Simonyan K, et al. The Kinetics Human Action Video datasets [J]. 2017.DOI: 10.48550/arXiv.1705.06950.

[5] Xu Long, Gong Chen, Yang Jie, et al. Violent video detection based on mosift feature and sparse coding [C] //2014 IEEE International Conference on Acoustics, Speech and Signal Processing, 2014:3538-3542.

[6] Febin I P, Jayasree K, Joy P T. Violence detection in videos for an intelligent surveillance system using MoBSIFT and movement filtering algorithm [J]. Pattern Analysis and Applications, 2020, 23(2):611-623.

[7] Sudhakaran S, Lanz O. Learning to Detect Violent Videos using Convolutional Long Short-Term Memory[C]. 2017 14th IEEE International Conference on Advanced Video and Signal Based Surveillance, 2017:33-34.

[8] Liang Qicheng, Li Yong, Yang Kaikai, et al. Long-term recurrent convolutional network violent Behaviour recognition with attention mechanism [J]. MATEC Web of Conferences, 2021, 336 (1): 5013.

[9] Feichtenhofer C, Fan Haoqi, Malik J, et al. SlowFast Networks for Video Recognition [C] //Proceedings of the IEEE/CVF international conference on computer vision. 2019: 6202-6211.

[10] Okan Köpüklü, Wei Xiangyu, Rigoll G. You Only Watch Once: A Unified CNN Architecture for Real-Time Spatiotemporal Action Localization [J]. arXiv preprint arXiv:1911. 06644, 2019.

[11] Li Hongchang, Wang Jing, Han Jianjun, et al. A novel multi-stream method for violent interaction detection using deep learning [J]. Measurement and Control, 2020, 53(5):796-806.

[12] Islam Z, Rukonuzzaman M, Ahmed R, et al. Efficient Two-Stream Network for Violence Detection Using Separable Convolutional LSTM [C] //2021 International Joint Conference on Neural Networks (IJCNN). IEEE, 2021: 1-8.

[13] Carreira J, Zisserman A Quo Vadis, Action Recognition? A New Model and the Kinetics datasets [J]. IEEE, 2017. DOI: 10. 1109/CVPR. 2017. 502.

[14] Direkoglu C. Abnormal Crowd Behavior Detection Using Motion Information Images and Convolutional Neural Networks [J]. IEEE Access, 2020, PP (99): 1-1. DOI: 10. 1109/ACCESS. 2020. 2990355.

[15] Dong Min, Fang Zhenglin, Li Yongfa, et al. AR3D: Attention Residual 3D Network for Human Action Recognition [J]. Sensors, 2021, 21(5):1656-1669.

[16] Li Zhan. Research on Video Violence Detection Algorithm Based on 3D Convolutional Neural Network [D]. Anhui University of Architecture, 2022. DOI: 10.27784/d. cnki. gahjz. 2022. 000160.

[17] XU Pengfei, ZHANG Pengchao, LIU Yaheng, et al. A human behavior detection algorithm based on SR3D network [J]. Computer Knowledge and Technology, 2022, 18(01):10-11. DOI: 10.14004/j.cnki.ckt.2022. 0068.

[18] Sanghyun Woo, Jongchan Park, Joon-Young Lee,In So Kweon. CBAM: Convolutional Block Attention Module. 2018.

[19] Wang C Y, Liao H Y M, Wu Y H, et al. CSPNet: A New Backbone that can Enhance Learning Capability of CNN [C] //2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW). IEEE, 2020. DOI: 10. 1109/CVPRW50498. 2020. 00203.

[20] Lim B, Ark S, Loeff N, et al. Temporal Fusion Transformers for interpretable multi-horizon time series forecasting [J]. International Journal of Forecasting, 2021(1). DOI: 10. 1016/j. ijforecast. 2021. 03. 012.

[21] Alwando E, Yie-Tarng Chen, Wen-Hsien. CNN-Based Multiple Path Searchfor Action Tube Detection in Videos [J]. IEEE Transactions on Circuits and Systems for Video Technology, 2018, 30 (1): 104-116.

[22] Wei Jiangchuan, Wang Hanli, Yi Yun, et al. P3D-CTN: Pseudo-3D Convolutional Tube Network for Spatio-Temporal Action Detection in Videos [C] //2019 IEEE International Conference on Image Processing (ICIP). IEEE, 2019: 300-304.

[23] Yang Xitong, Yang Xiaodong, Liu Mingyu, et al. STEP: Spatio-Temporal Progressive Learning for Video Action Detection [C] //Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. 2019: 264-272.

Cognitive Map Construction Based on Grid Representation

Yuxin Du
School of Computer Science and Engineering
Xi'an Technological University
Xi'an, China
E-mail: 1064139488@qq.com

Hongge Yao
School of Computer Science and Engineering
Xi'an Technological University
Xi'an, China
E-mail: 835092445@qq.com

***Abstract*—This paper investigates a grid-representation-based approach to spatial cognition for intelligent agents, aiming to develop an effective neural network model that simulates the functions of the olfactory cortex and hippocampus for spatial cognition and navigation. Despite progress made by existing models in simulating biological nervous system functions, issues such as model simplification, lack of biological similarity, and practical application challenges remain. To address these issues, this paper proposes a neural network model that integrates grid representation, reinforcement learning, and encoding/decoding techniques. The model forms a grid representation by simulating the integration of grid cells in the medial entorhinal cortex (MEC) with perceptual information from the lateral entorhinal cortex (LEC), which encodes and retains spatial location information. By leveraging attractor networks, convolutional neural networks (CNNs), and multilayer perceptrons (MLPs), the model achieves the storage of spatial location and environmental information, as well as the construction of cognitive maps. The experimental results show that after using this model, the map generation accuracy increased by 15%, the navigation accuracy of the agent in complex environments by 20%, and the target localization error was reduced to less than 10%, demonstrating a significant overall performance improvement in the grid-based cognitive map construction.**

Keywords-Grid representation; Cognitive map ;Infere-n-ce and generation

I. INTRODUCTION

Cognitive map is an internal structure in biological brain used to represent and understand environmental information, similar to the map in the city, which can guide people to find the destination [1]. In animal experiments, scientists have found that as animals move through their environment, the brain builds an internal cognitive map through interactions between neurons and synaptic adjustments that help animals find food, water, and more. The hippocampus and entorhinal cortex play key roles in memory [5]. The hippocampus is primarily responsible for short-term memory and spatial navigation, while the entorhinal cortex is involved in object recognition and spatial memory. Understanding the role of these two regions in memory and the relationship between them is crucial to understanding how brain-like cognitive maps are implemented.

The aim of the research on the construction of cognitive maps using artificial neural networks is to design a neural network model to simulate the learning and navigation abilities of humans and animals in the real world. This model can help people better understand how the human brain works, but also can provide new ideas and methods for the application of machine learning and artificial intelligence.

In the study of spatial cognition of agents, the construction method of cognitive map based on grid representation is an effective technical means [3]. By simulating the functions of the entorhinal cortex and hippocampus, agents can obtain spatial position, direction and target information, and use this information for spatial cognition, such as agent navigation and decision making [2]. This technique can not only improve the navigation accuracy and efficiency of the agent, but also enhance the adaptability and robustness of the agent in complex environments [4]. At the same time, it can be applied to robot navigation, autonomous vehicles, virtual reality and

augmented reality and other fi-elds to improve the intelligence level and adaptability of these systems.

II. Type Style and Fonts

A. VAE network

VAE works by sampling the potential space and then using a decoder to convert the sampled potential vector into a new data sample. Since the VAE's encoder maps the raw data onto a Gaussian distribution in the potential space, the potential vector can be obtained by sampling that Gaussian distribution. The sampled potential vector is then decoded to generate new data similar to the original data distribution.

One advantage of VAE in terms of augmentation of generated data is that it can control the degree of variation of generated data. By manipulating dimensions in different directions in the underlying space, selective changes can be made to t-he generated data. For example, you can interpolate or scale specific dimensions to generate data with specific properties or degrees of variation.

It is important to note that when applying VAE for data augmentation, it is necessary to ensure that the new data generated is similar to the distribution of the original data set, which can be achieved through the objective function when training the VAE model. VAE models are often trained with KL divergence of the underlying vector to minimize reconstruction errors, which ensures that the new data generated is similar to the original data set distribution, thus guaranteeing the quality of the new data generated

B. Reinforcement learning and agent cognition

The agent spatial cognition methods based on reinforcement learning usually employ end-to-end training, where action selection and path planning are performed directly from raw sensor readings [4]. Although this method can feed both dynamic and static obstacles into the network through a single frame, it is insufficient in the processing of interactive information. In order to solve this problem, some researchers use the received signal strength to define the return value and adopt Q learning method to complete path planning, but this method is not suitable for uncertain environments with a lot of dynamic obstacles, and the model is difficult to be transferred to the actual environment. Some researchers have trained UAVs by using Soft Actor-Critic (SAC) algorithm to perform autonomous obstacle avoidance in continuous action space using only image data, but the model has poor generalization ability and is difficult to adapt to the new environment [6]. This paper comprehensively considers the problem of multiple obstacles in complex and unfamiliar scenes, processes data through the method of simulated cognitive neuroscience, and further trains the representation vector obtained by grid representation to obtain the current position signal and target position signal of the cognitive map as the input of reinforcement learning network, optimizes the action selection of the agent, and improves the generalization ability of the model as a whole [7].

III. Network Model

A. Grid Representation

The grid representation model is a two-branch fusion model (as shown in Figure 1). Branch one extracts the key information of the current environment from the environment through the attractor network to simulate the visual information perceived by animals in the environment, and then extracts its features through the deformable convolution layer. Branch two obtains grid cells through the exploration of the environment by agents. Grid cells are believed to code the relatively unchanged spatial structure information that can span different environments and are input into the Hebbian competitive neural network for competition among position cells [8]. After multiple competitions, key position information is obtained, and then feature extraction is carried out through the deforming convolution layer. Feature fusion is performed on the features extracted from the double branches, and the final output is a kind of representation vector after fusion, which is called grid representation.

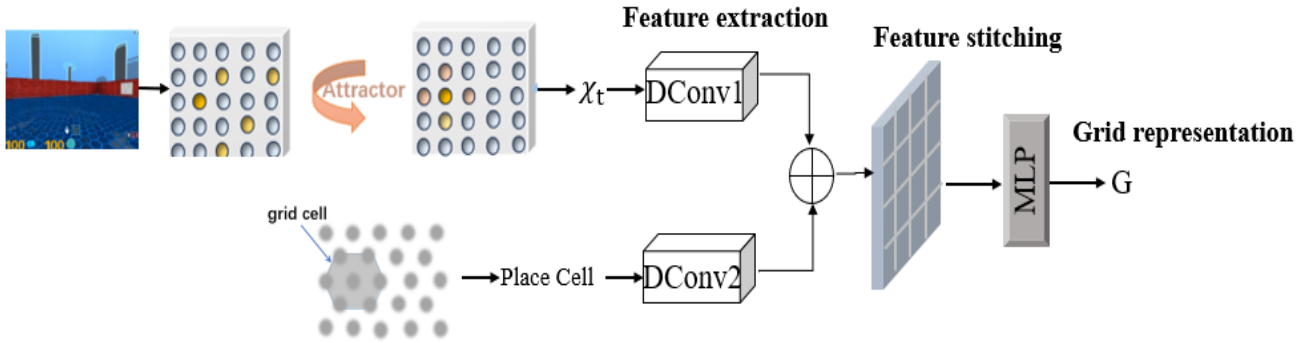


Figure 1. Grid representation model

The grid representation network model is mainly divided into the following parts:

- Visual perception branch: It is used to extract key information about the environment, similar to the visual information perceived by animals.
- Spatial coding branch: It is used to encode location information in the environment to obtain grid cells and location cells.
- Double-branch feature fusion: the features obtained from the visual perception branch and the spatial coding branch are fused to obtain the spliced feature vector.
- MLP training: Input the spliced feature vector into the multi-layer perceptron (MLP) for training, and finally output grid expression G.

The following is a step-by-step explanation of the composition and function of each part.

Branch of Visual Perception (Branch 1)， the branch of visual perception extracts key information about the current environment from the environment through the attractor network [9]. This branch uses the deformable convolution layer 1 to perform feature extraction on the extracted information to capture important visual features in the environment. The features obtained through this branch can capture the important visual features in the environment and provide the basis for the subsequent feature fusion.

Spatial Coding Branch (Branch 2)， the spatial coding branch explores the environment through agents and obtains grid cells. Grid cells are thought to encode information about spatial structures that can span different environments and remain relatively unchanged. The grid cells are entered into the position cells in the Hebb competitive neural network to make them compete with each other. After many competitions, get key position information. Then, the deformation convolution layer 2 is used to extract the features of these position information. The features obtained through this branch can capture the spatial structure information of the environment and provide the basis for the subsequent feature fusion.

Two-branch feature fusion is after obtaining the features of deformable convolution layer 1 and deformable convolution layer 2, double-branch feature fusion is carried out. The features obtained from the two branches are concatenated into a feature vector. This feature vector reflects key visual and spatial structure information in the environment, providing more comprehensive features for subsequent training.

MLP training is fusion vectors after processing are then processed by the multi-layer perceptron (MLP), and the vectors output by the activation function are linearly weighted and summed to obtain the final grid representation. The output result of this output layer is a high-dimensional vector representation G, reflecting the distribution and position information of the input information in space, as well as additional information after MLP processing. This output can be used as input to VAE (variational autoencoder) for further processing and encoding through inference models.

B. Inference and generation

The output of grid representation networks can be used as input to variational autoencoders (VAE). VAE is a generative model capable of learning potential representations of data and generating new data samples.

After the grid expression output G at step=t and the output gt-1 of the generated model at step= T-1 at the previous moment are combined as the input of VAE, the inference model of VAE will receive these two vectors as input (as shown in Figure 2) [10].

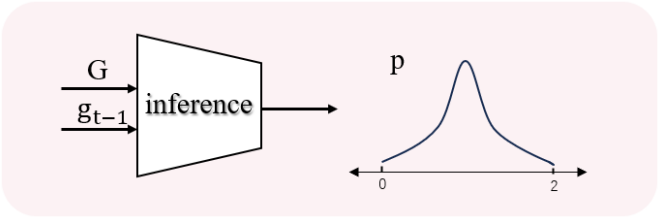


Figure 2. Inference model

Inference models map the input vectors G and gt-1 into the potential space using a nonlinear transformation function, a process that can usually be implemented by one or more neural network layers. The function learns a mapping relationship from the input space to the potential space so that the distribution of the input data in this space is as close as possible to its true potential distribution. In the potential space, the distribution of the data is modeled as a probability distribution, which is usually a multidimensional Gaussian distribution.

After obtaining a representation of the potential space, the inference model calculates the mean and variance of the potential variables. These parameters describe the distribution of potential Spaces and are used to generate new data samples in the decoder. This process is usually implemented using a regularization term, such as KL divergence or ELBO (Lower bound of evidence). The function of the regularization term is to make the potential distribution learned by the model as close as possible to a prior distribution, such as a Gaussian distribution. In order to ensure that the generated distribution can be as close as possible to the true posterior distribution of the task, we construct the variational lower as shown in equation (1), the Loss_Critic in the loss (2) function serves as a constraint for generating latent variables.

$$ELBO = E_T[E_{z_c \sim q_\varphi(z_c|g^T)}[R(T,z) + D_{KL}(q_\varphi(z_c \mid g^T) \| p(z_c))]] \quad (1)$$

$$Loss\_Encoder = \frac{1}{n}\sum_{i=1}^{n} D_{KL}(q_\varphi, p) + Loss\_Critic \quad (2)$$

Then sample a potential variable from the learned potential distribution. This latent variable mimics the place cell, or p, in the hippocampus, which represents spatial cognitive information. The sampling process is usually implemented using a reparameterization trick, by randomly sampling a sample from a potential distribution and then mapping this sample into the output space via a nonlinear transformation function.

In order to ensure the accuracy of the trained state distribution, a generation model is constructed. The input p is sampled from the posterior distribution q(p|g) obtained from the Encoder, and the output is the cognitive map reconstructed state feature g corresponding to the current p. We hope that the specific characteristics of the original state g can be restored according to the sampled p, and the accuracy of the generated p can be measured as a constraint term for generating potential variables.

In order to make the generated distribution as close as possible to the true posterior distribution of the state space, we construct the variational lower bound as in equation (3):

$$ELBO = E_T[E_{Zs \sim q_\phi(p_s|g)}[R(T,z_s) + D_{KL}(q_\phi(z_s \mid g) \| p(z_s))]] \quad (3)$$

In the decoding process, the output we want to get is the cognitive map g of the current moment. To achieve this, we need to add an output layer to the last layer of the decoder, which should be the same size as the dimensions of the cognitive map g. In the output layer, we can use nonlinear activation functions such as sigmoid function or softmax function to map the neuron's output to the range of [0,1] to get the final cognitive map g (as shown in Figure 3).

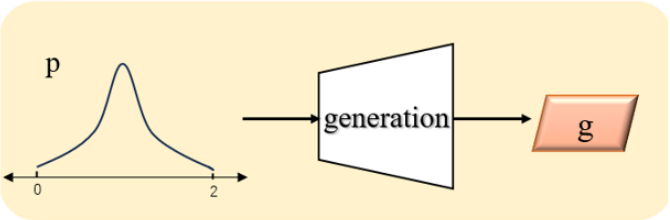


Figure 3. Generating the model

C. Cognitive map construction

The cognitive map construction network is divided into four parts: grid representation, inference and generation, target signal acquisition, and cognitive map construction.

Grid expression: In this part, the sensory perception in the memory is extracted through the attractor network, and the information obtained by grid discharge is used as input, feature extraction is carried out through convolution, and then feature fusion is carried out through MLP to obtain the fused feature vector as grid expression.

Inference and generation: In this part, the grid expression output and the output of the previous generation are taken as inputs together, and the input data is mapped to the potential space through inference to obtain the mean and variance of the potential variable, and then the distribution of the potential variable p. p sampled from the posterior distribution q(p|g) obtained by inference is taken as the input of generation, and after decoding by generation, g_t is output as the cognitive map reconstruction state feature corresponding to the current p.

Spatial cognitive information p is the core inferential information of cognitive map, and also an important basis for constructing cognitive map. Latent variables can be decoded to generate new cognitive maps based on these features.

Target signal acquisition: In order to extract walks from the environment and map them to the potential space as the inference input, the potential distribution p2 of the quadruple [s, a, r, s'] is obtained. The potential distribution contains the core feature distribution of the quadruple, which is goal code, that is, the target signal (as shown in Figure 4).

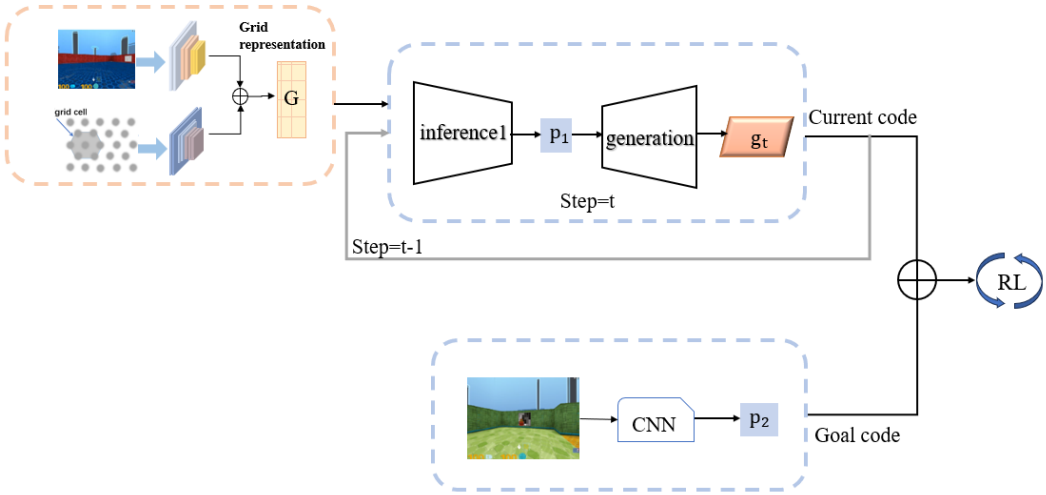


Figure 4. Overall model

IV. Experiment

Before you begin to format your paper, first write and save the content as a separate text file. Keep your text and graphic files separate until after the text has been formatted and styled. Do not use hard tabs, and limit use of hard returns to only one return at the end of a paragraph. Do not add any kind of pagination anywhere in the paper. Do not number text heads-the template will do that for you.

Finally, complete content and organizational editing before formatting. Please take note of the following items when proofreading spelling and grammar:

A. Data

In this experiment, a public data set containing road scenes is selected, which contains pictures of various road scenes and corresponding target annotation information, including vehicles, pedestrians and other targets. Such a data set can provide a variety of road environments and help evaluate the model's performance in different scenarios.

B. Navigation and obstacle avoidance performanc -e evaluation experiment

To verify the effectiveness of the cognitive map model in navigation and obstacle avoidance, we construct a virtual maze environment. The maze scene is a two-dimensional plane containing several static obstacles. The test object (the simulated agent) navigates the environment with the goal of getting from the start to the end while avoiding all obstacles.

Obstacle distribution: Obstacles are static and randomly distributed in different locations in the maze. Obstacles vary in shape and size to increase navigation difficulty and simulate diversity in a real environment.

Target point setting: set the starting point and end point of the maze, and require the tested object to reach the end point as quickly as possible without colliding with obstacles.

To visualize and analyze the performance of cognitive map models in navigation, we generate attention heat maps. The heat map shows the location and frequency distribution of the tested object throughout the maze.

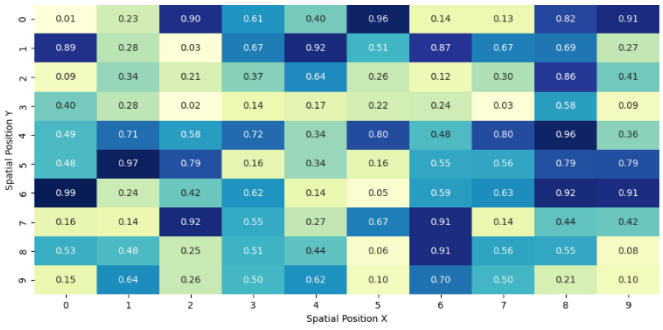


Figure 5. Location attention heat map

Through the heat map in the Figure 5, we can visually observe the position frequency distribution of the tested object in the maze. This heat map is generated by counting the occurrence frequency of the measured object in each spatial position (x, y). The darker the color, the higher the activity frequency of the measured object in this area; conversely, the lighter the color, the lower the activity frequency.

- High frequency areas:

In the heat map, certain areas (such as the (6, 0) and (0, 7) positions in the image) that show darker colors are hotspots of attention. This indicates that the subject stays in these positions for a long time or passes several times. The reason may be that these areas are critical turning points, bottleneck locations in the maze, or more complex path selection areas. These hot spots may have an important impact on navigation decisions, and the model may have performed more calculations or adjustments at these locations to choose the best path or avoid obstacles.

- Low frequency region:

Lighter colored areas indicate places where the subject passes less or spends less time. It may be because the path selection in these areas is relatively simple, or these areas are the edge of the maze or dead end, so the subject does not need to do too much stop in these areas.

The low-frequency activity in these regions shows how confident and efficient the model is on these partial paths, able to pass quickly.

From the distribution of the overall heat map, it can be seen that the behavior of the tested object is more concentrated in some specific areas, which may reflect the effectiveness of the path selection and obstacle avoidance strategy of the cognitive map model in this area. For example, in regions with more complex paths (such as the center or turning point of a maze), the color of the heat map changes more dramatically, reflecting that the model makes more path selection judgments in these regions.

To measure the accuracy of the model prediction, a heat map of the coincidence degree between the model prediction path and the actual navigation path is generated (as shown in Figure 6).

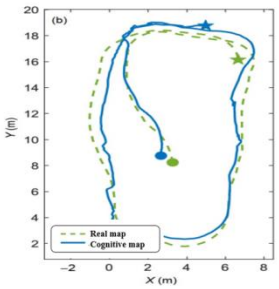


Figure 6. Trajectory comparison diagram

- Method: The predicted path of the model was compared with the actual path, and the areas with high coincidence between the two were marked.
- Objective: To analyze the predictive ability and execution ability of the model in different environments, and determine the difference between the path planning of the model and the actual execution.

C. Contrast experiment

In order to demonstrate the advantages of the cognitive map construction model based on grid representation design in the environment construction task, we compared the object recognition performance of different object detection models on the public data set and the cognitive map image generated by the cognitive map construction model. Common target detection models such as YOLOv5, Faster R-CNN and SSD were selected and deployed, trained and tested in a rigorous experimental environment to ensure the accuracy and reliability of the evaluation results. By comparing and analyzing the performance of the model in different scenarios, the purpose is to verify the data enhancement effect of the cognitive map building model, and explore its advantages and limitations in practical application.

In order to conduct comprehensive and effective data comparison, we choose the following common object detection models as data comparison models:

- YOLOv5: As a fast and accurate target detection model, YOLOv5 shows good performance and low computing cost.
- Faster R-CNN: Faster R-CNN is one of the classic target detection models in the industry, with high accuracy and strong robustness.
- SSD: SSD (Single Shot MultiBox Detector) is a target detection model that can detect multiple frames in a single time, with fast detection speed and high accuracy.
- DETR: DETR is an innovative target detection model with unique performance and advantages in the field of target detection.
- RT-DETR: RT-DETR is also an excellent target detection model with good adaptability to various scenarios.

These commonly used target detection models are selected as data comparison models, and each target detection model is deployed in the environment for training and testing to ensure the accuracy of the model effect.

This paper recorded average accuracy (mAP), accuracy (P), recall rate (R), F1 score and average accuracy (APcar) for each model.

In order to show the performance of each model on different data sets more intuitively, line charts for each index are drawn. Figure8 shows the comparison of the performance indicators of the target detection model under different data sets, and Figure9 shows the comparison of the execution rate of the target detection model under different data sets.

Finally, complete content and organizational editing before formatting. Please take note of the following items when proofreading spelling and grammar (Figure 7 and Figure 8):

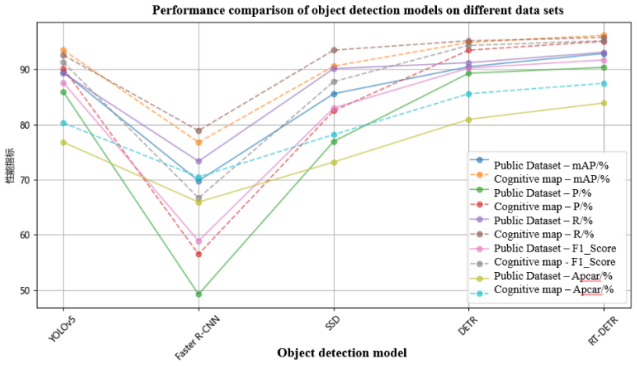


Figure 7. Display of target detection results

Figure 8. Execution rate diagram of target detection model

V. CONCLUSIONS

Based on the grid representation method, this paper discusses the performance of the agent in spatial cognition and navigation planning. By building a deep learning model that includes a grid representation generator, an encoder, a decoder, an inference model (including the hidden variable p), and a Goal Code generator, we have succeeded in providing the agent with a whole-process solution from environment awareness to target location to path planning.

Experimental results show that the model performs well in navigation and path planning tasks. Grid representation not only provides the agent with clear spatial structure information, but also enhances its ability to understand the layout of the environment. The introduction of hidden variable p enables the model to capture the main distribution of the environment, and improves the environmental adaptability and continuous decision-making ability of the model. The generation of Goal Code provides the agent with clear target guidance, which helps to generate accurate and efficient navigation path.

By combining deep learning techniques, we give the agent powerful spatial perception and decision-making capabilities, enabling it to navigate autonomously and complete tasks in complex environments. This not only provides strong support for the development of intelligent robots, autonomous driving and other fields, but also provides new ideas and methods for the research of future agents in the field of spatial cognition.

In the future, we will continue to optimize the model architecture and parameter Settings to improve the performance and generalization of the model. At the same time, we will also explore more advanced technologies and methods to promote the research and application of agents in the field of spatial cognition. We believe that in the near future, agents based on grid expression will be able to show their strong potential and value in more fields.

REFERENCES

[1] Whittington, J. C., McCaffary, D., Bakermans, J. J., & Behrens, T. E. How to build a cognitive map. Nature Neuroscience, 1-16. (2022).

[2] Farzanfar, D., Spiers, H. J., Moscovitch, M., & Rosenbaum, From cognitive maps to spatial schemas.Nature Reviews Neuroscience, R. S. (2022).

[3] Rueckemann, J. W., Sosa, M., Giocomo, L. M., & Buffalo, E. A. (2021). The grid code for ordered experience.Nature Reviews Neuroscience, 22(10), 637-649.

[4] Andrew Szot and Alex Clegg et al. Habitat 2.0: Training Home Assistants to Rearrangetheir Habitat, Advances in Neural Information Processing Systems, 2021.

[5] Foster, D., Morris, R., Dayan, P. et al. A model of hippocampally dependent navigation,using the temporal difference learning rule. Hippocampus 10, 1–16 (2021).

[6] The Tolman-Eichenbaum Machine: Unifying Spaceand Relational Memory through Generalization in the Hippocampal Formation. Whittington et al., 2020, Cell 183, 1249–1263.

[7] Everett M, Chen Y F, How J P. Collision avoidance in pedestrian-rich environments with deep reinforcement learning[J]. IEEE Access, 2021, 9: 10357-10377.

[8] Shanshan Qin, Shiva Farashahi, David Lipshutz, Coordinated drift of receptive fields in Hebbian/anti-Hebbian network models during noisy representation learning, Nature Neuroscience, pages339–349 (2023)

[9] Laura Cantini, Hope4Genes: a Hopfield-like class prediction algorithm for transcriptomic data, Scientific Reports, 337 (2019)

[10] Zhila Agharezaei, Reza Firouzi, Samira Hassanzadeh, Computer-aided diagnosis of keratoconus through VAE-augmented images using deep learning, Scientific Reports, 20586 (2023).

Research on the Financial Event Extraction Method Based on Fin-BERT

Jing He
School of Computer Science and Engineering
Xi'an Technological University
Xi'an, China
E-mail: hj15129793315@163.com

Yongyong Sun
School of Computer Science and Engineering
Xi'an Technological University
Xi'an, China
E-mail: sunyongyong@xatu.edu.com

***Abstract*—Event extraction is based on the event in the text as the subject information, based on the predefined event type and template, the structured event information is extracted, the existing event extraction model is mainly in the general domain, ignoring the prior knowledge in the domain and the dependency information between entities, and the existing methods do not address the problem of event theory dispersion and multiple events. In response to the above issues, this paper proposes a model based on Fin-Bert (Financial Bidirectional Encoder Representation from Transformers) and RATT (Relation-Augmented Attention Transformer). At the same time, this paper will make use of the structured self-attention mechanism to extract the dependencies between entities, use RAAT to fuse the dependency information between entities into sentence coding, and finally use the binary classification method to identify type of event and generate event records. Compared with the baseline method, the F1 value of the event extraction task on the ChFinAnn and Duee-fin datasets was improved by 2.5% and 2.8%, respectively.**

Keywords-Event Extraction; Relationship Information; Information Extraction; Natural Language Processing

I. INTRODUCTION

Event extraction takes the event information in the text as the subject information and extracts some specific information according to the predefined event types and templates to obtain structured data. Event extraction, deep learning, natural language processing, knowledge graph and other disciplines are interrelated, and the extracted information can be applied to the fields of public opinion control, the construction of knowledge graphs [1] intelligence collection [2] and information retrieval [3].

In the early days, event extraction was mainly done at the sentence level, which did not take into account the semantic information of the entire context. In order to solve the problem of sentence-level event extraction, we propose document-level event extraction. Nowadays, document-level event extraction is the mainstream method. However, there are two major problems: one is that document-level event contention is dispersed in each sentence, and the other is that there may be multi-event at a documentation.

The extraction of financial events can quickly extract structured financial events from many financial announcements to help investors understand the financial market and provide the most direct reference basis for investment transaction decisions. Nowadays, many document-level event extraction models are in general domain, and the texts in the financial field have their own unique professional characterization, and the existing models have poor extraction effects on financial texts. At the same time, these methods ignore the relationship information between entities, and cannot solve the question of dispersed event parameters and many events. In order to solve the shortcomings in the existing models, this paper studies the event extraction model in the field of Chinese finance, and proposes an event extraction method for Chinese financial corpus, which makes full use of the learning and expression ability of Fin-BERT (Financial Bidirectional Encoder Representation from Transformers), integrates prior knowledge in the financial field, and aims at the problem of scattered event parameters In this paper, a Fin-

BERT-based encoder and an event extraction method fusing entity relationship information Fin-RAAT are proposed, which use the dependency information and Financial information between entities to better extract financial documents with multiple events.

II. RELATED WORK

Early event extraction mainly focused on sentence granularity, extracting event information from the individual sentences of the text. Most of ways are difficult to extract to text scattered in the whole document event information, sentence level event extraction is difficult to extract to the event information in the whole document, so the document level event extraction by the attention of many scholars. Research methods also range from the beginning of pattern matching, machine learning to deep learning.

A. Event extraction method for pattern identification

Event extraction ways of pattern identification is mainly divided into two steps: pattern acquisition and pattern matching. Pattern acquisition is mainly to establish context constraints for the target text and then construct event templates, and pattern matching is based on the constructed event templates to match event information from the text, and the structure of matching depends on the construction of patterns.

Pattern matching event extraction methods based on the smaller datasets size requirements, high accuracy, good interpretability, can be well applied in the corresponding specific domains, but there are also obvious disadvantages: the first model construction relies on the domain expert's understanding of the text, and the effect of the extraction depends on the validity and completeness of the model. Second the model is very sensitive to domain knowledge and difficult to be quickly migrated to other domains. Thirdly, the extraction model is summarized from existing data sets by manual or weakly supervised algorithms, and the extraction model cannot deal with expressions that have not appeared in the data sets or are not representative. Meanwhile, the pattern matching method mainly focuses on the granularity of sentences, and it is difficult to cope with the situation where the event information is dispersed in different sentences by only relying on the pattern matching method.

B. Machine learning-based event extraction method

Based on the idea of machine learning event extraction method is the event extraction into classification of text fragments, first extract features in the construction model for event extraction, machine learning event extraction method to reduce the dependence on domain knowledge, but also reduce the construction mode of manpower cost, extraction effect, but it also has many shortcomings, first: machine learning method too dependent on the characteristics of manual construction, increase the workload of event extraction, it is also difficult to learn semantic information. Second: great deal high-quality annotated corpus, especially the trigger words in the event detection task. Third: machine learning methods are also stuck on the sentence granularity, and it is difficult to extract the event information scattered throughout the document.

C. Event extraction method based on deep learning

Deep learning based on event extraction methods are mainly used for event extraction by constructing multi-level neural network models. Neural networks can automatically obtain semantic features from the training corpus and has become the current mainstream approach. One class is to use recurrent neural network RNN, long short-term memory neural networks LSTM and Transformer [4] as the main models, which are used to get semantic representations of words and sentences at different granularity, but none of these methods can solve the problems of argument dispersion and multiple events well.

Du [5] et al. shown that document-level event extraction, while focusing on information of whole document, also focuses on the semantics at the sentence level. The thought is to transform event extraction assignment to a sequence notes assignment. Yang [6] et al. proposed the DCFEE method based on this idea, which starts from the

center sentence and then fuses the sentence-level information and document-level information granularity by means of elemental complementation. The method starts from the center sentence and then fuses the two granularity of sentence rank and document rank by means of element complementation to complete document-rank event extraction, which avoids the disadvantage of not being able to obtain complete event information from a sentence.

Zheng [7] et al. proposed an end-to-end event extraction method Doc2EDAG. Some variants based on Doc2EDAG have emerged, for example. Xu [8] et al. proposed a model GIT using a tracker. Zhu [9] et al. improved the Doc2EDA and GIT approaches by proposing a lightweight model, PTPCG, which senses events through pseudo-trigger information and adopt complete graph pruning and decoding to generate event records, the method uses LSTM coding layer to fuse the knowledge of the corpus, then the combination of parameters of each event is considered as a complete graph before pruning, and a non-autoregressive model decoding strategy is adopted for the event's ontology extraction, and the problems in the Doc2EDAG and GIT methods are well solved by the pruning strategy of changing the graph.

All of the above ways have good extraction effect, but there are still some problems. The DCFEE method models Chinese financial event extraction assignment at document level, and detection of multiple events relies on the trigger annotation, and the extraction effect is poor in the face of unlabeled triggers, and the three models, Doc2EDAG, GIT, and PTPCG, are all general-purpose domain event extraction models, which do not take into account the financial domain's related knowledge, relationships between entities are not taken into account, therefore, the extraction effect in the financial sector is not good.

Fin-RAAT model construction

The model framework of Fin-RAAT ways proposed in this paper is displayed in Figure 1. Fin-RAAT is an end-to-end event extraction model, namely entity extraction and representation (EER), document relationship extraction (DRE), entity and sentence coding (ESE), and event record generation (ERG). In the process of training and reasoning, rather than only through pipeline connection. Meanwhile, the model uses relationship dependency information. Furthermore, a relationship prediction task was increased to the model to adequately exploit relational message between entities to enhance the accuracy of event extraction. The essay will introduce these four modules in detail.

D. Entity extraction and representation

In this paper, entity extraction is regarded as a sequence labeling task, given a document D that has much sentences $\{s_1, s_2 ... s_i\}$, using the transformer encoder for present all token sequence. In this paper, Fin-Bert encoder is used to present the toke sequence. To classify the token using Conditional Random Field (CRF), the vector representation of the token is transformed into a labeled entity, we use the classical BIOSE sequence labeling paradigm, where the label is predicted to be predicted by the following calculation (1), and then, all the intermediate embedded mentions of the entity are extracted and the sentences are connected into a matrix. The loss function for named entity recognition is defined to calculation (2).

$$\hat{y}_{ne} = CRF(Trans(D)) \tag{1}$$

$$L_{ne} = -\sum_{si \in D} \log P(y_i \mid s_i) \tag{2}$$

The s_i represents the 1th sentence in the document. y_i represents correct sequence annotation.

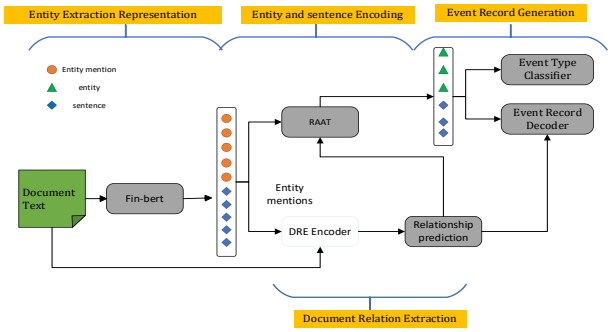


Figure 1. Model framework

With the rapid development of internet technology, the capability of computational processing of large volumes of text has also been increasing rapidly. This has made the contextual environment of text more complex, and the understanding of contextual semantics by computers directly affects the final outcomes. Consequently, pre-trained models based on language models are being increasingly utilized, among which BERT is one of the most widely used training models. This model was proposed by Google in 2018 and has achieved state-of-the-art results in various tasks within the field of Natural Language Processing (NLP). The BERT model is composed of multiple stacked bidirectional Transformer encoders, which capture the contextual semantic features of text from two directions, thereby better representing natural language text. In practical applications, there is no need to make extensive changes to the structure of specific tasks; instead, by introducing a new output layer and optimizing BERT's performance, advanced models can be established for different tasks. This approach is also referred to as pre-training and fine-tuning.

The Fin-Bert encoder used in our paper is a pre-training model to the Chinese financial domain based on the BERT [10] architecture open-sourced by Entropy Simple Technology AI Lab. The native Chinese BERT input is sliced and diced at the granularity of a word, which doesn't take into account the relationship between co-occurring words or phrases in the domain, and thus fails to learn the implicit prior knowledge in the domain and reduces the model's learning effect. Fin-Bert is trained using full word masks, and the training data are mainly from the financial domain, including financial and financial news, announcements of listed companies and financial encyclopedia entries. The Fin-Bert model takes into account the relationship between Chinese characters and words, which is in line with the linguistic habits of the Chinese language, and permits the model to study the a priori knowledge within financial domain implied between phrases, which makes its extraction in the financial domain better than that of other models. The training model of Fin-Bert is shown in Figure 2.

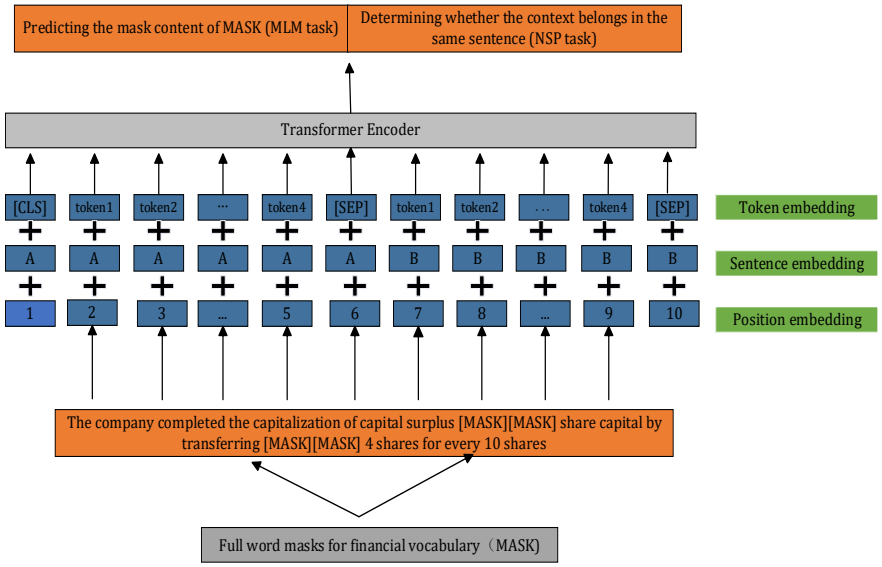


Figure 2. The trained model of Fin-Bert

E. Document relationship extraction

This paper defines the structure between two kinds of entities:

1) Co-occurrence structure: Entities appear in the same sentence.

2) Two entity references refer to the same entity.

Both structural associations can be present between entities individually or simultaneously. Therefore, this paper has four relationships.

The entities can have both structural associations, either individually or simultaneously. Thus, there are 4 total relational dependencies between entities. The document relationship extraction part utilizes document text D and entity $\{e_1, e_2 \ldots\ldots e_j\}$ as data inputs, and outputs the relationship of entities $\{[e_1^h, e_1^t, r_1], [e_2^h, e_2^t, r_2] \ldots\ldots [e_k^h, e_k^t, r_k]\}$ pair as the kth ternary, with entities, tail entities and relations represented within the ternary. In order to predict the dependencies between event parameters, structured self-attention network is used for relationship prediction between entities, in this paper, normal cross entropy is used to predict the labels of each entity pair only, and the type of relationship is hypothesized through the following function equation (3):

$$\hat{y}_{i,j} = \arg\max(e_i^T W_r e_j) \tag{3}$$

Where e_i refers to the entity embedding from the encoder model document relational extraction, and W_r denotes biaffine matrix trained by DRE task. The loss function is shown below:

$$L_{\text{dre}} = -\sum_{y_{i,j} \in Y} \log P(y_{i,j} \mid D) \tag{4}$$

Where $y_{i,j}$ represents the correct relation label between entities i and j, and D represents all relation pairs of an entity.

F. Entity and sentence encoding

Now embedding the entity mentions and sentences from the Entity Extraction and Representation component and the list of predicted ternary relationships from the Document Relationship Extraction component, the section then encodes the above data and outputs an embedding that is effectively integrated with the relationship information. In this part, this paper will transform the triples into computable matrices and use the RAAT structure to encode the entity and sentence correlations efficiently, RAAT has a unique attention computation module, which has two parts: self-attention and relationship-enhanced attention computation, and through the alteration module, the dependencies between the entities are integrated efficiently into the document encoding. The RAAT structure is shown in Figure 3.

G. Event record generation

With the output of the previous section, entities and sentence embedding, this event record generation module usually contains two parts: event type classification and event record decoding, given a sentence embedding, for each event type several binary predictions are used to determine whether the corresponding event is recognized or not, and if any of the classifiers recognizes the event kind, then event record decoder is activated to iteratively generating each parameter of the corresponding event type. The loss function is shown below:

$$L_{\text{pred}} = -\sum_i \log(P(y_i \mid S)) \tag{5}$$

The y_i represents the label of the ith event type, and $y_i = 1$ if the event type is the event record of i then $y_i = 1$ else $y_i = 0$, S represents input embedding of sentences.

An EDAG is a series of iterations whose length is equal to the number of roles of a particular type, where the goal every iteration is to forecast the affair parameters of a particular affair character, and the input to each iteration contains embedding of entities and sentences, and the input predicted parameters become part of the input to the next iteration. However, unlike EDAG, this paper employs RAAT instead of the original transformer part in the iteration process, EDAG employs a memory structure to record the extracted parameters and adds a role-type representation to predict the current iteration parameters, however, it is difficult for this method to capture the dependencies between memorized entities, candidate arguments, and sentences, and employs

RAAT instead of the original transformer. The RAAT structure can connect entities and candidate parameters in memory by means of relational triples extracted from the relational extraction section, and a structure representing dependencies can be constructed. Before predicting the event parameters for the current iteration, the matrix T is constructed so that the dependencies can be integrated into the attention computation, and after extracting the parameters and adding them to memory, the new matrix T is generated to accommodate the forecast for next iteration. The RAAT here has the same structure as the RAAT at sentence entity encoding, but the parameters are independent and the event record decoder loss function is defined:

$$L_a = -\sum_{v \in V_D} \sum_{e} \log(P(y_e \mid (v, s))) \tag{6}$$

The V_D , v represents the set of nodes and D represents the event parameters of the extracted event records so far s denotes the embedding of the sentence and candidate event parameters and y_e denotes the label of argument candidate e in current step. $y_e = 1$ means that e is the basic event parameter corresponding to the current step event role, else $y_e = 0$.

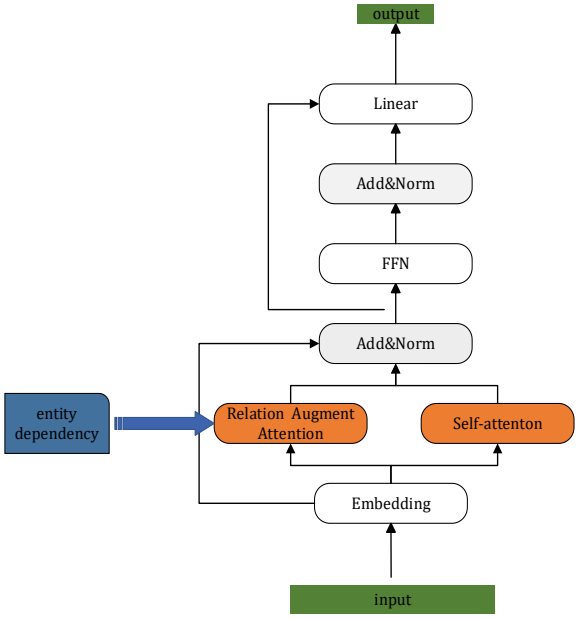


Figure 3. RAAT structure

III. EXPERIMENTS

A. Experimental datasets

This paper use ChFinAnn [7] and Duee-fin [11] as the datasets. The basic information of the datasets is displayed in Table1.

The ChFinAnn, which contains 32,040 documents, including 9,306 multi-event documents, accounting for about 29% of the total. 98% of the event thesauri are dispersed in different sentences, with the average of 20 sentences contained in each text, the longest text contains about 6,200 Chinese characters. The datasets contain five event types.

The Duee-fin datasets are published by Baidu, Inc. and the results are obtained using an online review in the experiment. The datasets event types are derived from common financial events, 13 in total, which are company listing, equity reduction, shareholder increase, corporate buyback, corporate financing, share repurchase, share pledge, release of pledge, corporate bankruptcy, loss, being interviewed, winning a bid, and executive change. The datasets contain a total of 92 event-theoretic meta-role types.

TABLE I. DATASETS INFORMATION

	Training data	Validation data	Test data	Event types
ChFinAnn	25632	3204	3204	5
Duee-fin	7015	1171	About 3500	13

B. Experimental design

The experimental environment is displayed in Table 2.

TABLE II. EXPERIMENTAL ENVIROMENT

Hardware Information	Configure
Operating system	Windows 10
Internal storage	32GB
CPU	AMD R9-5900HX
Video card	RTX-3080-LAPTOP
Memory	16GB
Exploitation environment	Python 3.7

The following experimental setup was used for this experiment: the same vocabulary as BERT was used for the participle table, the Fin-BERT encoder was used with the base size of the BERT architecture (dh = 768, dl = 12), a stochastic initialization was used for the parameters to be trained in the model.

C. Experimental results

In this experiment, several baseline algorithms will be selected to compare with this paper's model as a way to display the effectiveness of our paper's model, using the Precision(P), Recall(R), and F1 value(F1) are used as evaluation indexes.

Table 3 demonstrates experimental dismissal of each baseline algorithm of Fin-RAAT and Doc2EDAG, GIT, and PTPCG on the ChFinAnn datasets, and the results prove that the P, R, and F1 value of this paper's model have been improved in comparison with each baseline algorithm.

TABLE III. EXPERIMENTAL EXTRAXTION OF CHFINANN DATASETS

	P	R	F1
Doc2EDAG	80.3	70.5	77.5
GIT	82.3	78.4	80.3
PTPCG	88.2	69.1	79.4
Fin-RAAT	84.0	79.9	81.9

Table 4 shows the event extraction results of various baseline algorithms with Fin-RAAT on the Duee-fin datasets.

TABLE IV. DUEE-FIN DATASETS

	Dev			Online test		
	P	R	F1	P	R	F1
Doc2EDAG	73.7	59.8	66.0	67.1	51.3	58.1
GIT	75.4	61.4	67.7	70.3	46.0	55.6
PTPCG	71.0	61.7	66.0	66.7	54.6	60.0
Fin-RAAT	76.1	71.3	73.0	70.3	56.1	62.8

Analyzing the individual metrics, the F1 value of this paper's model is improved by 6.7%, which is much better than other models. Meanwhile, for the online test evaluation, this paper's model shows a significant increase of 2.8% in F1 score over the baseline. This result display that this paper's model outperforms existing methods.

Table 5 represents the comparison of the extraction results of the baseline model and the model of this paper on the ChFinAnn datasets in terms of single and multiple events, Single refers to a document in the datasets that contains only 1 event, Multi refers to a document that contains more than one event, and All refers to all the data in the datasets. The results show that Fin-RAAT model is better than other models in extracting multiple events.

TABLE V. COMPARISON OF EVENT EXTRACTION EXPERIMENT RESULTS IN CHFINANN DATASETS

	F1		
	Single	Multi	All
Doc2EDAG	81.0	67.4	77.5
GIT	87.6	72.3	80.3
PTPCG	88.2	691	79.4
Fin-RAAT	87.9	75.3	81.9

Figure 4 compares the F1 values of the event extraction results of this paper's method with the baseline model on the ChFinAnn datasets for five types of events, and it can be intuitively seen that for most of the event type. The financial event extraction model proposed in this paper has better extraction results on financial texts.

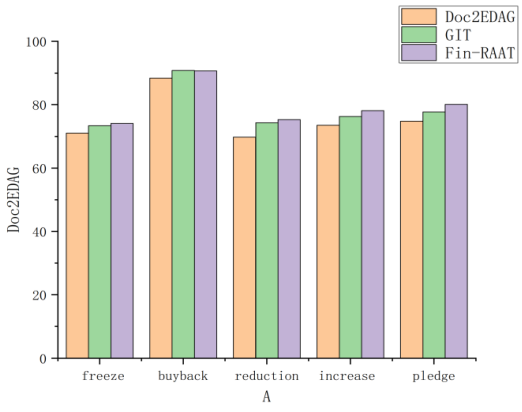


Figure 4. Types of Events Extracted Resulting F1 Values

IV. CONCLUSIONS

In this paper, we propose a financial event extraction model Fin-RAAT based on Fin-BERT as an encoder and fusion of dependencies between entities, which firstly inputs the input documents into the Fin-BERT layer in terms of words for encoding and entity recognition. After obtaining the entity mentions, the structural relationship dependencies between entities are predicted, and the relationship dependencies are fused into the document encoding through RAAT, followed by event type detection through binary classification, and after obtaining the event kinds, the event records are generated basis pre-defined event templates. The experimental results demonstrate that the financial event extraction model proposed in this paper can well solve the problems of event thesis element dispersion and multiple events in document event extraction. The method achieves good results on two benchmark datasets, but there are also areas for improvement in changing the model, the model only considers the structural dependence between entities, and does not consider the causal relationship between entities, and the model is larger, the training cost is high, in the future, we can make the most of this relationship information between entities to improve results of financial event extraction.

REFERENCES

[1] BOSSELUT A, LE BRAS R, CHOI Y. Dynamic neuro symbolic knowledge graph construction for zero-shot commonsense question answering. Proceedings of the AAAI conference on Artificial Intelligence. 2021, 35(6): 4923-4931.

[2] LI M. ZHU Y, WANG R. An Empirical Study on Utilizing Neural Network for Event Information Retrieval. International Conference on Computer Science and Communication Technology, 2020, 1621(1): 51-56.

[3] LI M. ZAREIAN A, LIN Y, et al. GAIA: A fine-grained multimedia knowledge extraction system. Proceeding of the 58th Annual Meeting of the Association for Computational Linguistics: System Demonstrations, 2020: 77-86. K Elissa, “Title of paper if known,” unpublished.

[4] WADDEN D, WENNBERG U, LUAN Y, et al. Entity, relation, and event extraction with contextualized span representations. Proceeding of 2019 Conference on Empirical Methods in Natural Language Processing and 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP). 2019: 5784-5789.

[5] DU X, CARDIE C. Document-level event role filler extraction using multi-granularity contextualized encoding. Proceedings of Association for Computational Linguistics (ACL). 2020: 634-644.

[6] YANG H, CHEN Y, LIU K, et al. DCFEE: A document-level Chinese financial event extraction system based on automatically labeled training data. Proceeding of Association for Computational Linguistics (ACL). Melbourne, Australia, 2020: 50-55.

[7] ZHENG S, CAO W, XU W, et al. Doc2EDAG: An end-to-end document-level framework for Chinese financial event extraction. Proceeding of 2019 Conference on Empirical Methods in Natural Language Processing and 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP), 2019: 337-346.

[8] XU R, LIU T, LI L, et al. Document-level event extraction via heterogeneous graph-based interaction model with a tracker. Proceeding of the Joint Conference of the 59th Annual Meeting of the Association for Computational Linguistics and the 10th International Joint Conference on Natural Language Processing (ACL-IJCNLP). 2021: 3533-3546.

[9] ZHU T, QU X, CHEN W L, et al. Efficient Document-level Event Extraction via Pseudo-Trigger- aware Pruned Complete Graph. Proceedings of Internation Joint Conference on Artificial Intelligence. 2022: 4552-4558.

[10] DEVLIN J, CHANG M W, LEE K, et al. BERT: Pre-training of deep bidirectional transformers for language understanding. Proceeding of Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (NAACL-HLT).2019: 4171-4186.

[11] LI, X. et al. Duee-fin: A document-level event extraction dataset in the financial domain released by baidu. (2021) [2023-04-06]. https://aistudio. baidu.com/ competition/detail/46.

Research on Construction Site Safety Q&A System Based on BERT

Ang Li
School of Computer Science and Engineering
Xi'an Technological University
Xi'an, China
E-mail: la630670659@163.com

Jianguo Wang*
Research Institute of Artificial Intelligence and Data Science
Xi'an Technological University
Xi'an, China
E-mail: wjg_xit@126.com

***Abstract*—This paper aims to utilize the pre-trained language model BERT from deep learning to construct A question and answer system specifically targeting safety knowledge in construction sites, thereby enhancing safety management on-site and increasing workers' awareness of safety issues. Through extensive reading of literature related to construction site safety and the integration of practical case studies, this research compares various pre-trained language models such as word2vec, Pre-trained RNN, GPT, and BERT, analyzing their respective advantages and disadvantages. Despite the fact that word embedding methods such as word2vec have improved the effectiveness of natural language processing to some extent, their ability to understand context is limited. Pre-trained RNNs, although capable of handling sequential data, suffer from the problem of gradient disappearance when dealing with long-range dependencies. In contrast, the GPT model performs well in generative tasks; however, due to its reliance on a unidirectional language model, it falls short in understanding bidirectional contexts. Ultimately, it was determined that a method based on BERT would be most suitable for improving the model to meet the safety needs of construction sites. The system can accurately understand and respond to safety-related questions posed by workers, thereby preventing accidents and ensuring the safety of construction site personnel. This study not only explores the optimization and adjustment of the BERT model but also evaluates its performance in practical application scenarios, providing new technological means for safety education and management within the construction industry.**

Keywords-component; BERT; Question and Answer System; Construction Industry; LLM

I. INTRODUCTION

With the rapid advancement of computer science and artificial intelligence technologies, the field of Natural Language Processing (NLP) has made groundbreaking progress, particularly in the development of intelligent question answering systems. These systems aim to mimic human communication by understanding questions posed by users in natural language and then providing precise answers, significantly enhancing the efficiency and quality of information retrieval. Moreover, this technology demonstrates extensive application potential across various fields, especially in the realm of construction safety. Faced with complex and dispersed safety knowledge and business data, traditional management methods are no longer sufficient to meet the demands for efficient and accurate queries.

Currently, safety management on construction sites faces severe challenges. The site environment is dynamic, with numerous safety hazards present. Additionally, safety standards and operational guidelines are scattered across various sources, such as paper documents, websites, and internal databases. This not only increases the difficulty of information retrieval but also significantly affects the efficiency of safety education, training, and daily management. Therefore, the development of an intelligent question answering system that can integrate and rapidly parse these unstructured data sources has become particularly urgent.

Alongside the development of computer technology, question answering systems have seen extensive development across various vertical domains. The first chatbot, Eliza [1], was invented by foreign computer scientists in 1966 and was

used in psychological counseling, utilizing matching rules similar to decision trees to analyze input sentences. Although question answering systems based on pattern matching and manually written rules could provide reasonable responses, building a large number of dialogue templates to satisfy the rich expressiveness of language was required, making this approach applicable only in a few specialized fields. In recent years, with the advancement of deep learning, the use of neural networks for lexical analysis has become a focal point of research. Currently, in the industry, deep learning-based text matching mainly includes three categories: vector similarity-based matching, deep neural network-based matching, and matching based on pre-trained models [2]. The NNLM [3] proposed a method of computing text vector similarity primarily to address the synonym problem in traditional statistical methods; however, the training resource consumption is significant, requiring several weeks of training using 40 CPUs for a dataset with millions of entries [4]. With the expansion of natural language processing applications, researchers have combined deep learning models for NLP tasks [5], mainly to solve semantic representation at the sentence level and asymmetry problems in text matching. Microsoft introduced DSSM [6] in 2013, which was the earliest deep semantic matching model. The DSSM model maps Queries and Docs to a low-dimensional semantic space and measures the relevance between Query and Doc through Cosine similarity. Although interaction models can capture deeper semantic information with more complex neural network structures, they overlook syntactic and global information across sentences. In 2018, Google introduced the pre-training model BERT (Bidirectional Encoder Representations from Transformers), which ranked first in various NLP task leaderboards [7]. BERT provides a new approach for the rapid acquisition of safety knowledge on construction sites. Leveraging its strong contextual understanding and generation capabilities, BERT has achieved leading positions in multiple NLP tasks. The model, through unsupervised pre-training on large-scale corpora, has learned rich linguistic features and complex semantic relationships, demonstrating excellent performance during the fine-tuning phase for specific tasks.

In summary, the work presented in this paper will focus on utilizing the latest deep learning technologies, particularly BERT and subsequent pre-trained models, in conjunction with domain knowledge and the requirements of real-world applications, to research and develop an efficient and accurate question answering system for construction site safety knowledge. This endeavor aims to improve the current efficiency of safety management and enhance the overall safety levels of construction sites. Through literature analysis and technical exploration, this study hopes to provide strong technological support for the informatization and intelligent transformation of the construction industry.

II. IMPROVED BERT MODEL

A. BERT Model Analysis

In traditional question answering systems, static word vector algorithms such as Word2Vec are often employed. These methods map input words into unique and invariant vectors, resulting in word embeddings that do not incorporate contextual information and cannot resolve the issue of polysemy in texts. In recent years, the emergence of pre-trained language models has ushered in a new era for natural language processing, replacing the original static word vectors and downstream task integration, thereby enhancing performance. BERT is currently one of the most successfully applied language models in the industry. Due to its powerful feature representation capabilities, fine-tuning with BERT requires only a small amount of relevant corpus data and appropriate parameter adjustments to reach the usability threshold of the domain [11]. To meet the needs of generating responses according to specific question-answer styles in a construction site safety knowledge QA system, the BERT model has been introduced. BERT is a deep bidirectional pre-trained language model based on the Transformer architecture [12], capable of capturing the common features of both preceding and following contexts to encode unlabeled text. When used in different natural language

processing scenarios, BERT typically requires the addition of an extra input layer. It is currently the most popular and widely used pre-trained language model, serving as the foundational architecture for enhancing model performance across various downstream tasks.

Prior to BERT, the most successful pre-trained language model was GPT, which utilized a left-to-right autoregressive approach for pre-training. However, in terms of language understanding, it is considered a bidirectional process because information from both the left and right sides of the current content is helpful for comprehension. In other words, GPT represents a unidirectional process, for reasons including the need for a defined direction. Naturally, this approach breaks down long texts into smaller segments and generates them progressively, essentially decomposing the process. Secondly, if all the content were fed into a bidirectional model, it could lead to information leakage. Figures 1 and 2 illustrate the unidirectional and bidirectional processes of GPT.

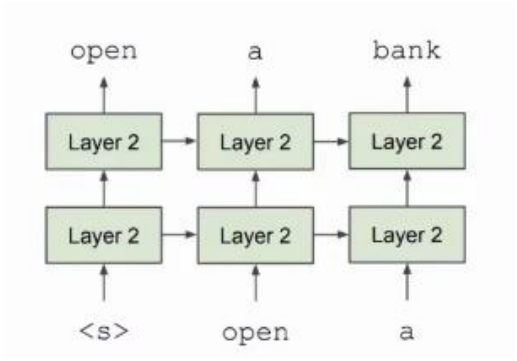


Figure 1. Unidirectional context build representation incrementally.

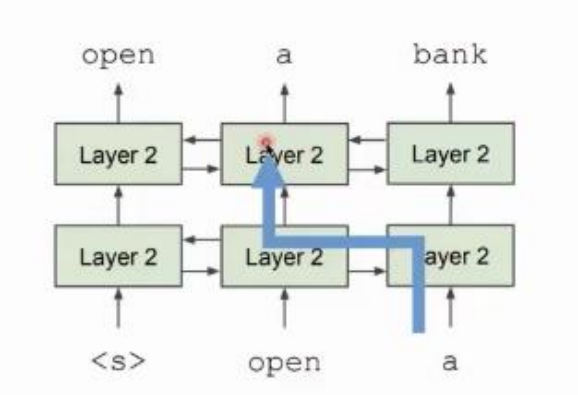


Figure 2. Bidirectional context words can "see themselves".

BERT proposes a solution with the masked language model approach. For example, in the sentence 'the man went to the [MASK] to buy a [MASK] of milk,' two [MASK] tokens cover the original words 'store' and 'gallon.' This masked language model is the core pre-training task of BERT, similar to a cloze test process. It employs a strategy of randomly masking 15% of the tokens. The choice of 15% is a trade-off. There are two main considerations: if the masked proportion is too low, there would be very little supervisory signal; if the proportion is too high, there would be very little usable information left in the text.

Masking solves the problem of information leakage, but it also introduces another issue: masked tokens do not appear during downstream tasks, creating a significant discrepancy between the pre-training and fine-tuning stages, which may degrade the model's performance.

B. Improvement Ideas and Methods

Therefore, to address the aforementioned issues, this paper proposes the following: when masking 15%, it should be divided into several subtypes for processing. Specifically, 80% of the time, the tokens should be replaced with [MASK]. For example, 'went to the store' would be converted to 'went to the [MASK],' meaning that the word 'store' is replaced with [MASK].

Next, 10% of the time, randomly replace the token with another word at a 10% chance. For example, in 'went to the store,' the word 'store' would be replaced with 'running,' and the model would be required to predict 'store' from 'running.' This approach forces the model to pay attention to words that do not appear to be masked, thus maintaining a better representation.

However, there is still another issue: the model might always assume that the randomly inserted word in the real scenario is incorrect. To address this, an additional strategy is implemented. An additional 10% of the time, the word order is kept normal. For example, 'went to the store' remains 'went to the store,' meaning that 'store' is used to predict 'store.' By combining these three masking strategies—masking, random replacement, and

keeping the original word—the issues brought about by masking are mitigated.

The structure of BERT is illustrated in Figure 4.4. It consists of three main components: the input layer, the encoding layer, and the output layer. For the input layer, the input representation vectors are composed of word embeddings, position embeddings, and segment embeddings. Segment embeddings are used to distinguish between different sentences in the conversation of the question answering system, and both the segment embeddings and position embeddings require learning by the model. The special token [CLS] (classification token) is placed at the beginning of the sequence to serve as a classifier. After passing through the final Transformer layer, this classification token aggregates the representation information of the entire sequence. The [SEP] token acts as a separator placed between two sentences. The specific input representation of BERT is detailed in Figure 3.

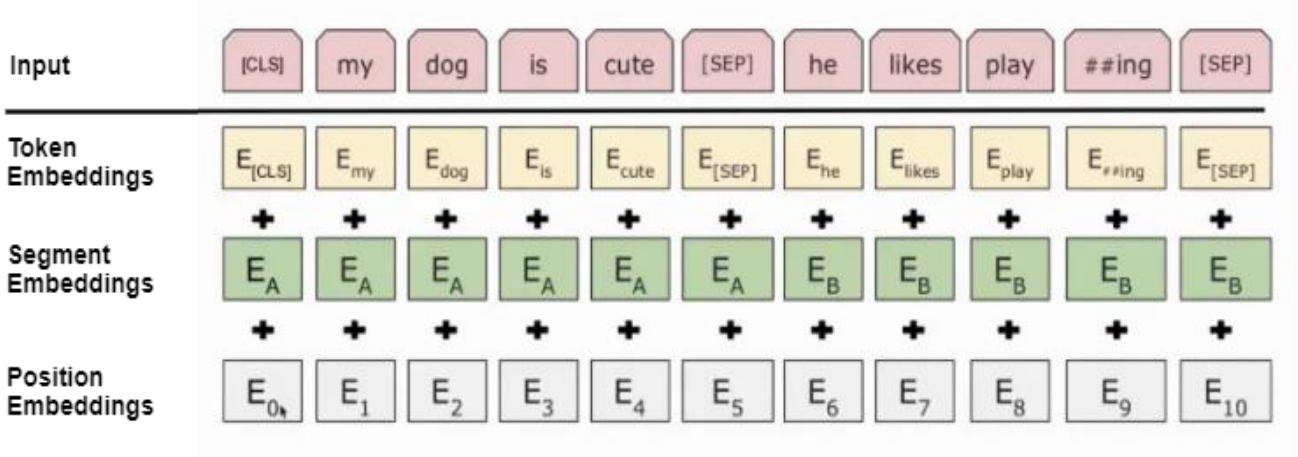


Figure 3. BERT model input diagram.

C. Model architecture

The construction site safety knowledge question answering system based on BERT has a model architecture that primarily includes an input layer, an Embedding layer, a Transformer Encoder, and an output layer. The structure is shown in Figure 4. The input layer receives preprocessed question inputs, with special [CLS] and [SEP] tokens added before the questions. In the Embedding layer, each token is mapped to a high-dimensional vector space, incorporating a combination of Token Embedding, Segment Embedding, and Position Embedding. The Transformer Encoder is the core component of BERT, consisting of multiple identical layers of multi-head self-attention (Multi-Head Attention) and feed-forward neural networks (Feed-Forward Neural Networks). This layer is responsible for capturing the semantic dependencies and structural features within the text. For the output layer, the output vector at the [CLS] position is taken as the representation of the entire sequence, followed by a fully connected layer (Fully Connected Layer) and a SoftMax activation function to predict the start and end positions of the best answer.

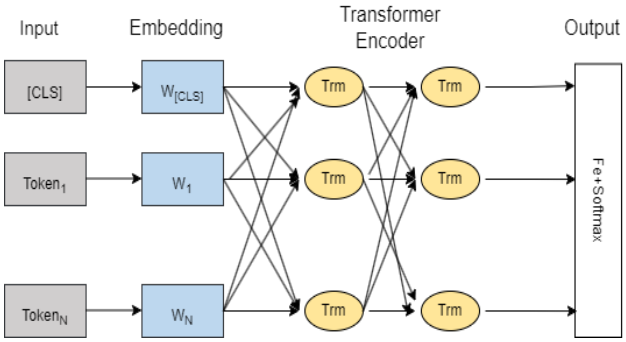


Figure 4. Model architecture of BERT-based Q&A system.

III. DATA SET CONSTRUCTION AND PRE-PROCESSING

A. Data set construction

Firstly, the objective of constructing the dataset for this paper is to encompass a broad and specific range of construction site safety knowledge,

ensuring diversity and practicality in the question-answer pairs. Data was collected using web crawlers from professional websites, forums, and blogs in the construction industry, focusing on safety questions and their answers that workers might encounter in real-life situations. In addition, common question-answer pairs were obtained from journal articles, e-books, and legal regulations related to construction safety. For the collected dataset, the model only requires the pure conversational information from the corpus. Therefore, text cleaning was performed to remove irrelevant characters, punctuation marks, unify the text format, and categorize the original materials into different safety topics to facilitate the creation of targeted question-answer pairs. The final number of items in the dataset is shown in TABLE I.

TABLE I. NUMBER OF SAMPLES IN THE DATA SET

Data Set Number	**Dataset Name**		
	training set	***validation set***	***test set***
67700	47750	9975	9975

B. Pre-processing steps

Preprocessing is a crucial step in this project, directly impacting the model's training effectiveness and the ultimate performance of the question answering system. Initially, tokenization is required, which involves breaking continuous text into individual words or lexical units. This is particularly important for Chinese text, as there are no clear word boundaries. In this paper, we utilize the Full-Tokenizer that comes with the BERT model. The process involves first applying Basic-Tokenizer to obtain a relatively coarse list of tokens, followed by Word-Piece Tokenizer to achieve the final tokenization results. Subsequently, the tokenized text needs to be converted into Tokens from the model's vocabulary. For words not found in the vocabulary, the UNK (unknown) Token is used, or sub-word encoding is applied to transform the text into a numerical form that the model can interpret.

Wherever Times is specified, Times Roman or Times New Roman may be used. If neither is available on your word processor, please use the font closest in appearance to Times. Avoid using bit-mapped fonts if possible. True-Type 1 or Open Type fonts are preferred. Please embed symbol fonts, as well, for math, etc.

IV. EXPERIMENTS AND ANALYSES

A. Experimental environment

The model in this study is implemented in Python and utilizes the deep learning framework PyTorch. The training process was conducted under the Ubuntu operating system. To ensure stable operation and good performance of the model, we meticulously configured the software and hardware environment for our experiments. The specific environmental configurations are detailed in TABLE II, including the operating system, version of the programming language, version of the deep learning framework, and versions of other necessary software libraries.

Additionally, we specified the GPU model and memory capacity to ensure adequate computational resources during the model training process. Through carefully configured experimental settings, we were able to effectively manage and optimize the training process, thereby ensuring the smooth progress of our research.

TABLE II. EXPERIMENTAL ENVIRONMENT CONFIGURATION PARAMETERS

Experimental Environment	**Configure**
operating system	Ubuntu
development language	Python3.8.8
development framework	Pytorch1.8.0
CPU	Intel(R) Core(TM) i7-8750H CPU @ 2.20GHz2.21 GHz
GPU	NVIDIA GeForce GTX3070Ti 8G
random access memory (RAM)	Kingston 2400Mhz 16.0 GB

B. Experimental parameter settings

For the training of the designed question answering model, the learning rate was set to 10e-5, the batch size was set to 16, and the number of epochs for the training set was set to 20. The

Adam optimizer was used for optimization, and a redesigned loss function based on cos (u, v) was adopted, as shown in Equation (1). Here, t ∈ {0, 1} indicates whether the samples are similar, where u and v represent the sentence feature vectors of Question 1 and Question 2, respectively. The purpose of the loss function is to maximize the similarity for positive sample pairs and minimize the similarity for negative sample pairs.

$$L = t \cdot \left(1 - cos(u,v)\right) + (1-t) \cdot \left(1 + cos(u,v)\right) \quad (1)$$

The experimental training parameters are set as shown in TABLE III:

TABLE III. EXPERIMENTAL ENVIRONMENT CONFIGURATION PARAMETERS

Experimental Parameters	Retrieve A Value
Learning rate	2e-5
Batch Size	16
Num of epoch	20
Length of Maxseq	128

C. Evaluation indicators

This paper adopts commonly used evaluation metrics in deep learning, namely accuracy, recall, and the F1 score. Accuracy represents the ratio of correctly predicted positive samples to all positive samples. The F1 score is the harmonic mean of precision and recall. Recall represents the ratio of correctly predicted positive samples to all actual positive samples. The formulas for these three metrics are given in Equations (2), (3), and (4):

$$P_{precision} = \frac{TP}{TP + FP} \quad (2)$$

$$r_{recall} = \frac{TP}{TP + FN} \quad (3)$$

$$F = 2 * \frac{r_{Recall} * P_{precision}}{r_{Recall} + P_{precision}} \quad (4)$$

In the above equations: TP represents the number of true positives, where the model correctly predicts positive samples as positive; TN represents the number of true negatives, where the model correctly predicts negative samples as negative; FP represents the number of false positives, where the model incorrectly predicts negative samples as positive; FN represents the number of false negatives, where the model incorrectly predicts positive samples as negative.

D. Experimental results and analyses

For this study, to measure the accuracy of the BERT-based question answering system, it was compared with several baseline models, and the results are shown in TABLE IV:

TABLE IV. COMPARATIVE EFFECTS OF DIFFERENT BASELINE MODELS

Modelling	Evaluation Metrics		
	P/%	***R/%***	***F1/%***
LSTM	72.72	68.63	70.61
Text-CNN	73.40	70.23	71.78
BERT	80.5	80.96	81.65

According to TABLE IV, it can be observed that the LSTM and Text-CNN models exhibit comparable performance on the question answering task. However, the pre-trained BERT model demonstrates significantly better results than the other baseline models. Furthermore, during the training process, BERT's performance tends to improve with the increase in model size, as illustrated in Figure 5. This suggests that larger variants of BERT are more capable of capturing complex patterns in the data, which leads to higher accuracy in answering questions related to construction site safety. The superior performance of BERT can be attributed to its ability to understand context and semantic relationships within the text, which is critical for accurately interpreting the nuances of safety-related queries.

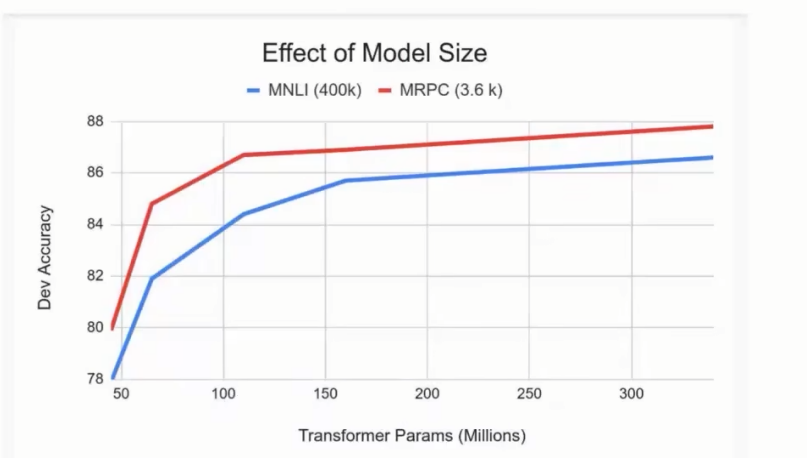


Figure 5. Effect of BERT with increasing model size.

V. WEB PAGE DESIGN

A simple graphical user interface (GUI) was constructed based on the model trained in this study, designed to facilitate user interaction. The entire web page project was created using the Next.js template configuration within WebStorm. The web design encompasses components such as a toolbar panel, button definitions and implementations, dialog lists, dialog messages, and dialogue handling.

Initially, a small sidebar was designed in this paper, which influenced the overall structure definition and page routing management of the interface. Two buttons were placed on the left sidebar: one for chat and another for role selection. The chat button facilitates dialogue handling, while the role button allows users to choose from various scenarios, including roles such as engineer, project manager, worker, etc. The coding of the webpage was carried out using an object-oriented approach, defining functions, methods, properties, and incorporating packages similar to how it is done in Java.

Subsequently, button functionality was introduced to achieve zooming effects on the page, enhancing the web UI's infrastructure modules. Although implementing a single button might appear minor, the inclusion of multiple buttons necessitated the design of a generic button framework along with configuration storage. When operating these buttons, configurations could be set and utilized, thereby adjusting the interface scale. Buttons serve as a small feature point that leads to the realization of various modules within the overall web UI architecture, encompassing aspects such as button definition, CSS design, and TypeScript syntax.

Following this, a dialog module was added to implement a list of dialog boxes. Corresponding test data was also included, and new sessions were created upon clicking the "+" button. Building upon the completed sidebar implementation, the development of the dialog list window was initiated. When users engage in dialogue with the model, different roles may emerge, such as engineer, project manager, or worker. These windows need to be displayed in a list format within the dialog box list, akin to how conversations are presented with different contacts in WeChat. Consideration was given to what information should be displayed, such as who was being chatted with, the number of chat messages exchanged, and when the last conversation ended, and how to present this information on the interface.

Subsequently, the focus was placed on implementing the dialog message feature, including the setup of corresponding sub-routing, page navigation, message transmission, interface design, and realization. After establishing the dialog box list, the content of the dialog box message panel needed to be realized. This meant that when a user clicked on an element in the list of dialog boxes, a corresponding dialog box message would appear on the right side, along

with an input field for messages. Additionally, the dialog box list was stored locally within the browser to preserve the user's conversation history. Such information could also be retrieved via server-side APIs. However, relying more on the browser's local storage rather than server storage reduces server load, particularly beneficial for non-corporate users who wish to deploy such services with minimal dependencies. Similarly, message information was stored locally in the user's browser, eliminating the need for server-side storage. Future expansions might include server-side storage options. Finally, concerning message transmission and presentation, considering that the data returned by the model is in Markdown format, especially for code snippets, rendering this data becomes necessary for better readability. Thus, extending Markdown rendering capabilities was essential.

For the of dialogue roles, this paper predefines these roles within the program, allowing users to initiate a dialogue operation with a specific role by simply clicking on it. Additionally, clicking the role button navigates users to a list of roles. Each role introduces its own functionalities, and once a user chooses to converse with a particular role, a chat message is created in the user's dialog box.

The web system's user interface described above is illustrated in Figure 6.

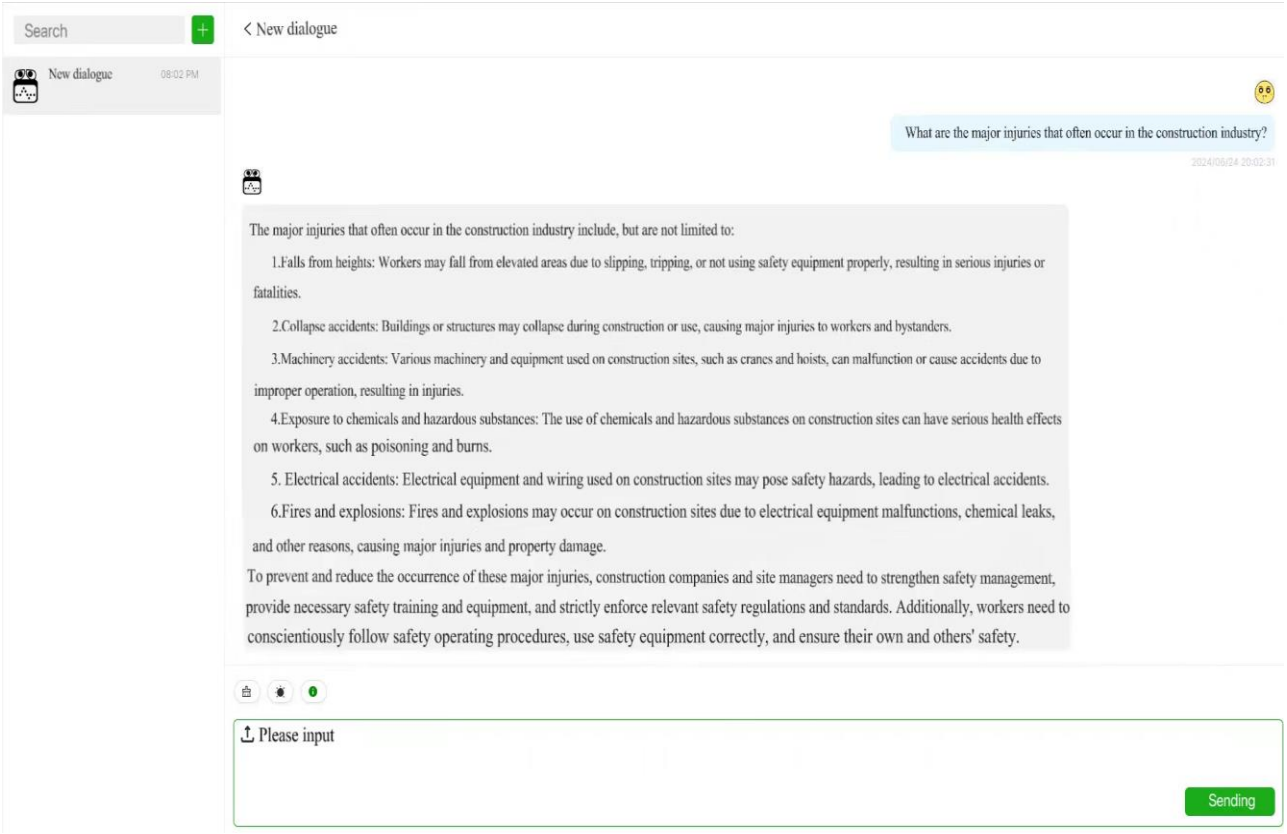


Figure 6. Graphical interface of the construction site Q&A system.

VI. CONCLUSIONS

The primary contribution of this paper is the proposal of a construction site safety knowledge question answering system based on the BERT model. Reliable safety knowledge from the construction industry was gathered from the internet to build a testing dataset, and experiments were conducted on this dataset with the proposed model. Comparisons with other baseline models demonstrated that this model can be effectively applied in the construction site industry. The system enables workers to quickly and accurately acquire safety knowledge, with a deeper understanding of the textual queries provided by users, resulting in more precise answers and

significantly improving the efficiency and accuracy of safety information retrieval for workers.

However, the limitations of the BERT model in terms of computational resource consumption, domain-specific knowledge constraints, handling of long texts, and interpretability cannot be ignored. The BERT model requires substantial computational resources for training and may lack flexibility when dealing with domain-specific knowledge. Additionally, its performance on long texts is inferior to that on short texts, and the model's decision-making process lacks transparency and interpretability. Therefore, future research will explore efficient question answering retrieval model architectures and algorithm implementations to further enhance the question answering capabilities of the system and improve the performance of its responses.

REFERENCES

[1] Love, Rachel et al. "Natural Language Communication with a Teachable Agent", CoRR (2022).

[2] Qian Yangge et al. "A review of deep learning-based text semantic matching." Software Guide 21.12 (2022): 252-261.

[3] Sun, Simeng, and Mohit Iyyer. "Revisiting Simple Neural Probabilistic Language Models", North American Chapter of the Association for Computational Linguistics abs/2104.03474 (2021): 5181-5188.

[4] Zhang, Min, and Li, J. T." Generative pre-training model." Chinese Science Foundation 35.03 (2021): 403-406. doi: 10.16262/j.cnki.1000-8217.2021.03.014.

[5] Jingsheng Zhao, et al. "A study of text representation in natural language processing." Journal of Software 33.01 (2022): 102-128. doi:10.13328/j.cnki.jos.006304.

[6] Zeng, Yanhong et al. "Learning Pyramid-Context Encoder Network for High-Quality Image Inpainting", 2019 IEEE/CVF CONFERENCE ON COMPUTER VISION AND PATTERN RECOGNITION (CVPR 2019) abs/1904.07475 (2019): 1486-1494.

[7] Linyang, Li et al. "BERT-ATTACK: Adversarial Attack Against BERT Using BERT", Conference on Empirical Methods in Natural Language Processing 2020.emnlp-main (2020): 6193-6202.

[8] Lee, Jinhyuk et al. "BioBERT: a pre-trained biomedical language representation model for biomedical text mining", Bioinformatics 36.4 (2020): 1234-1240.

[9] Zhang, Zhengyan et al. "Ernie: Enhanced Language Representation with Informative Entities", 57TH ANNUAL MEETING OF THE ASSOCIATION FOR COMPUTATIONAL LINGUISTICS (ACL 2019) abs/1905.07129 (2019): 1441-1451:

[10] Subakan, Cem et al. "Attention is All You Need in Speech Separation", IEEE International Conference on Acoustics, Speech, and Signal Processing abs/2010.13154 (2021): 21-25.

[11] Sufeng, Duan, and Zhao Hai. "Attention Is All You Need for Chinese Word Segmentation", Conference on Empirical Methods in Natural Language Processing 2020.emnlp-main (2020): 3862-3872.

A Novel Variance Reduction Proximal Stochastic Newton Algorithm for Large-Scale Machine Learning Optimization

Dr.Mohammed Moyed Ahmed
ECE Department
JNTUH, Hyderabad, INDIA
E-mail: mmoyed@gmail.com

***Abstract*—This paper introduces the Variance Reduction Proximal Stochastic Newton Algorithm (SNVR) for solving composite optimization problems in machine learning, specifically minimizing F(w) + Ω(w), where F is a smooth convex function and Ω is a non-smooth convex regularizer. SNVR combines variance reduction techniques with the proximal Newton method to achieve faster convergence while handling non-smooth regularizers. Theoretical analysis establishes that SNVR achieves linear convergence under standard assumptions, outperforming existing methods in terms of iteration complexity. Experimental results on the "heart" dataset (N=600, d=13) demonstrate SNVR's superior performance: Convergence speed: SNVR reaches optimal solution in 5 iterations, compared to 14 for ProxSVRG, and >20 for proxSGD and ProxGD. Solution quality: SNVR achieves an optimal objective function value of 0.1919, matching ProxSVRG, and outperforming proxSGD (0.1940) and ProxGD (0.2148). Efficiency: SNVR shows a 10.5% reduction in objective function value within the first two iterations. These results indicate that SNVR offers significant improvements in both convergence speed (180-300% faster) and solution quality (up to 11.9% better) compared to existing methods, making it a valuable tool for large-scale machine learning optimization tasks.**

Keywords-Composite optimization; Machine learning; Stochastic Newton method; Variance reduction; Convergence analysis

I. INTRODUCTION

In recent years, the field of machine learning has seen a surge in the importance of composite optimization problems. These problems, characterized by the sum of a smooth convex function and a non-smooth convex regularizer, arise in various applications ranging from regression models to classification tasks. The challenges in solving such problems are twofold. First, the large number of samples leads to high computational costs in calculating function values and gradients. Second, the optimization often occurs in high-dimensional spaces, further complicating the process [1-3].

Traditional approaches to solving these problems have evolved significantly over time, from full gradient descent methods to more sophisticated stochastic and variance-reduced algorithms [4-5]. Despite these advancements, there remains a need for algorithms that can more effectively balance computational efficiency with convergence speed, especially in the context of large-scale machine learning problems with non-smooth regularizers [6-7].

In this paper, we introduce a novel algorithm : the Variance Reduction Proximal Stochastic Newton Algorithm (SNVR). SNVR combines the strengths of variance reduction techniques with the proximal Newton method, offering a powerful new approach to solving composite optimization problems. Our algorithm builds upon the ideas of stochastic average gradient methods but incorporates them into a proximal Newton framework.

The SNVR algorithm offers several key advantages: 1. It leverages the fast convergence properties of Newton-type methods. 2. It incorporates variance reduction techniques to mitigate the noise inherent in stochastic methods.

3. It maintains the ability to handle non-smooth regularizers through the use of proximal operators.

The remainder of this paper is organized as follows: Section 2 presents a literature review of recent advancements in optimization algorithms for machine learning. Section 3 describes the SNVR algorithm in detail. Section 4 presents the theoretical analysis and convergence properties of SNVR. Section 5 provides numerical results demonstrating the algorithm's performance on real-world datasets. Finally, Section 6 concludes the paper and discusses potential future research directions.

II. LITERATURE RESEARCH

Recent years have seen significant advancements in optimization algorithms for large-scale machine learning problems. This section provides an overview of key developments within the last five years, focusing on stochastic methods, variance reduction techniques, and quasi-Newton approaches.

Stochastic Quasi-Newton Methods, Guo et al. (2023) [8] provided a comprehensive overview of stochastic quasi-Newton methods for large-scale machine learning. Their work highlighted the importance of balancing convergence speed, computational cost, and memory usage. The authors emphasized the need for further research into developing more efficient and scalable stochastic quasi-Newton methods, particularly for high-dimensional problems. Convergence Analysis for Non - strongly Convex Functions, Zhang et al. (2020) [9] made significant progress in understanding the convergence properties of Stochastic Gradient Descent (SGD) for non-strongly convex smooth optimization problems. Their novel analysis proved that SGD can achieve linear convergence under specific conditions, establishing a connection between the smoothness of the objective function and the convergence rate. This work provided valuable insights into the behavior of SGD in a broader class of optimization problems.

Variance Reduction Techniques Variance reduction has emerged as a crucial approach for improving the efficiency of stochastic optimization methods. Sinha et al. (2021) [10] conducted a comprehensive review of various variance reduction techniques used in deep learning. Their work discussed the strengths and weaknesses of each method, including their applicability, computational complexity, and impact on convergence. This review serves as a valuable guide for researchers and practitioners in selecting appropriate variance reduction techniques for specific deep learning tasks. Asynchronous Parallel Methods, As the scale of machine learning problems continues to grow, asynchronous parallel optimization methods have gained attention. Qianqian et al. (2021) [11] proposed an asynchronous parallel stochastic quasi-Newton method that combines the benefits of quasi-Newton updates with asynchronous parallel processing. Their approach leverages a novel communication mechanism to ensure consistency and stability in parameter updates across multiple processors, resulting in significant speedups in training times compared to traditional methods.

Block Coordinate Descent with Variance Reduction, Gower et al. (2018) [12] introduced a new variance reduction technique for Stochastic Block Coordinate Descent (SBCD) methods. Their approach significantly reduces the variance in gradient estimates, achieving convergence rates comparable to full gradient methods. Y. Chen et al. [13] addresses the challenge of optimizing non-convex functions, which frequently arise in machine learning tasks such as deep learning. This work offers substantial improvements in efficiency and scalability for large-scale optimization problems, particularly those with block structure.

These recent advancements in optimization algorithms for machine learning have paved the way for more efficient and scalable methods. However, there is still room for improvement, particularly in developing algorithms that can effectively handle non-smooth regularizers while maintaining fast convergence rates. Our proposed Variance Reduction Proximal Stochastic Newton Algorithm (SNVR) aims to address these challenges by combining the strengths of variance reduction techniques with the proximal Newton method.

III. MATHEMATICAL MODEL

A. Problem Formulation

These papers consider the following composite optimization problem:

$$\min_{w\in\mathbb{R}^d} = F(w)+\Omega(w)=\sum_{i=1}^{N} f_i(w)+\Omega(w) \quad (1)$$

$f(x)$ is a smooth convex function composed of N individual smooth convex functions $f_i(w)(i=1,2,...,N)$

- $\Omega(w)$ is a convex, potentially non-smooth regularization function.
- $w\in\mathbb{R}_d$ is the parameter vector to be optimized.

This formulation encompasses a wide range of machine learning problems, including regularized least squares, logistic regression, and support vector machines.

B. Assumptions

1) The component functions $f_i(.)$ are strongly convex, and their gradient functions satisfy the L-Lipschitz condition.

2) The Hessian matrix $\nabla^2 f_i(w)$ is bounded for any non-empty subset S.

$$\begin{aligned}&\mu\,||\,v-w\,||\\ &\le\ f_i(v)-f_i(w)-\nabla f_i(w)^T(v-w)\\ &\le||\,v-w\,||\end{aligned} \quad (2)$$

Here $\mu>0$ and $L>0$ are constants.

The Hessian matrix $\nabla^2 f_i(w)$ is bounded for any non-empty subset S. Specifically, there exist constants λ_1 and λ_2 such that,

$$\lambda_1 I_d \le \nabla^2 f_i(w) \le \lambda_2 I_d \quad (3)$$

C. Key Lemmas

Lemma 3.1

Let w_* be the optimal solution of the problem (1). Then.

$$\begin{aligned}&\mathbb{E}\,||\,\nabla F(w_k)-\nabla F(w)\,||^2\\ &\le 4L[\phi(w_k)-\phi(w_*)+\phi(w_{k+1})+\phi(w_*)]\end{aligned} \quad (4)$$

Lemma 3.2

Let $\phi(w)=F(w)+\Omega(w)$, and assume that $\nabla F(w)$ is L-Lipschitz continuous. Let $w_{k+1}=prox_\alpha^H(w_k-\alpha H^{-1}g_k)$, where $g_k=\nabla F(w)$, α is the step size, and $0<\alpha\le 1/L$. Then,

$$\begin{aligned}\phi(w_k)\ge\phi(w_{k+1})+g_k^T H(w_k-w_{k+1})\\ +\Delta(w_{k+1},w_k)+\frac{1}{2}||\,g_k\,||^2_{H^{-1}}\end{aligned} \quad (5)$$

Where,

$$\begin{aligned}\Delta(w_{k+1},w_k)=\Omega(w_{k+1})-\Omega(w_k)\\ -\nabla\Omega(w_k)^T(w_{k+1}-w_k)\end{aligned} \quad (6)$$

D. Main Convergence Theorem:

Let $w_*=\arg\min_w\phi(w)$, $0<\alpha\le 16\lambda_1/\lambda_2^2$, and assume that the assumptions in Section 3 hold. Then,

$$\mathbb{E}[\phi(w_{k+1})-\phi(w_*)]\le\rho^*[\phi(w_k)-\phi(w_*)] \quad (7)$$

Where $\rho^*=(1+\frac{7L\alpha\lambda_2}{\lambda_1})<1$.

Let:

$$\begin{cases}w=w_k, w_+=w_{k+1}, v=v_k, g=g_k, u=w_*\\ H=H_k^{-1}, \Delta=\Delta(w_{k+1},w_k)\\ \breve{w}_{k+1}=prox_\alpha^H(w_k-\alpha H_k^{-1}\nabla F(w)\end{cases} \quad (8)$$

Then by Lemma 3.1 we can obtain:

$$\begin{aligned}&\mathbb{E}\,\|\breve{w}_{k+1}-w_*\|_{H_k}^2\\&\le \mathbb{E}\,\|w_k-w_*\|_{H_k}^2+2\alpha\mathbb{E}[\phi(w_{k+1})-\phi(w_*)]\\&+L\alpha^2\lambda_2^2[\phi(w_k)-\phi(w_*)+\phi(w_{k+1})-\phi(w_*)]\end{aligned}\quad(9)$$

$\mathbb{E}\Delta(w_{k+1},w_k)=0$, we have:

$$\begin{aligned}&\mathbb{E}\,\|w_{k+1}-w_*\|_{H_k}^2\\&\le \mathbb{E}\,\|w_k-w_*\|_{H_k}^2+2\alpha\mathbb{E}[\phi(w_{k+1})-\phi(w_*)]\\&+8L\alpha^2\lambda_2^2[\phi(w_k)-\phi(w_*)+\phi(w_{k+1})-\phi(w_*)]\end{aligned}\quad(10)$$

By strong convexity, we know:

$$\begin{aligned}&\mathbb{E}\,\|\breve{w}_{k+1}-w_*\|_{H_k}^2\\&\le \mathbb{E}\,\|w_k-w_*\|_{H_k}^2+2\alpha\mathbb{E}[\phi(w_{k+1})-\phi(w_*)]\\&+16L\alpha^2\lambda_2^2[\phi(w_k)-\phi(w_*)]\end{aligned}\quad(11)$$

Therefore, these works have:

$$\begin{aligned}&\mathbb{E}[\phi(w_{k+1})-\phi(w_*)]\\&\le\left(1+\frac{7L\alpha\lambda_2}{\lambda_1}\right)[\phi(w_k)-\phi(w_*)]\end{aligned}\quad(12)$$

Since $0<\alpha\le\frac{16\lambda_1}{\lambda_2^2}$, This work have $\rho^*=(1+\frac{7L\alpha\lambda_2}{\lambda_1})<1$ which implies that the SNVR algorithm converges linearly.

IV. ALGORITHM IMPLEMENTATION

The Variance Reduction Proximal Stochastic Newton Algorithm (SNVR) represents a sophisticated approach to solving large-scale machine learning optimization problems. At its core, SNVR combines the strengths of variance reduction techniques with the power of Newton's method, all while maintaining the ability to handle non-smooth regularizers through proximal operations.

The algorithm begins with an initialization phase, where an initial point is chosen, and key parameters such as batch size, convergence threshold, and step size are set. A crucial step in this phase is the computation and storage of the full gradient and the inverse of the Hessian matrix at the initial point. This forms the foundation for the subsequent iterative process.

The heart of SNVR lies in its main loop, where the algorithm iteratively refines the solution until convergence. In each iteration, a subset of the data is randomly selected, enabling the algorithm to work efficiently with large datasets. This stochastic approach is key to the algorithm's scalability.

A distinguishing feature of SNVR is its use of a variance-reduced gradient estimate. By maintaining a memory of previously computed gradients and updating only a subset in each iteration, SNVR achieves lower variance in its gradient estimates compared to standard stochastic gradient methods. This variance reduction technique is crucial for the algorithm's stability and fast convergence.

Algorithm 1 Stochastic Newton Variance Reduced (SNVR) Algorithm

Require: Initial point w_0, batch size b, tolerance ϵ, learning rate α, maximum iterations M

Ensure: Optimal solution w_*

Initialize $k = 0$

Calculate and store the gradients $\nabla f_1(w_0), \nabla f_2(w_0), ..., \nabla f_N(w_0)$

Compute the Hessian matrix H at w_0 and its inverse H^{-1}

Calculate $w_1 = prox_\alpha^H(w_0 - \alpha H^{-1}\nabla f(w_0))$

Set $k = 1$

while $|\phi(w_{k+1})-\phi(w_k)|>\in|$ **do**

Randomly select a subset S_k of size b from the set {1, 2, ... , N }

Compute the gradients $\nabla f_i(w_k)$ for $i \in S_k$

Calculate $v_k = \nabla f_{s_1}(w_k) - \nabla f_{s_1}(w_{k-1}) + \nabla f(w_{k-1})$

Compute the Hessian matrix H at w_k and its inverse H^{-1}

Calculate $w_{k+1} = prox_\alpha^H(w_k - \alpha_k^{-1}H^{-1}v_{k-1})$

Update the gradients $\nabla f_i(w_{k+1})$ for the updated subset S_k

Set $k = k + 1$

end while

return w_{k+1}

The algorithm then computes the Hessian matrix at the current point, incorporating second-

order information into the optimization process. This Newton-type update allows SNVR to make more informed steps towards the optimal solution, particularly beneficial in regions where the objective function has high curvature.

The parameter update step employs a proximal operator, which is essential for handling the non-smooth regularizer term in the objective function. This operator allows SNVR to effectively navigate the optimization landscape even in the presence of non-differentiable components.

After each update, the algorithm checks for convergence based on the change in the parameter values. This process continues until the algorithm converges or reaches a maximum number of iterations.

The SNVR algorithm's unique combination of variance reduction, Newton-type updates, and proximal operations positions it as a powerful tool for tackling complex optimization problems in machine learning.

V. EXPERIMENTAL PROCESS AND RESULTS

To validate the theoretical properties and assess the practical performance of the SNVR algorithm, we conducted a comprehensive set of numerical experiments. Our study focused on a regularized least squares problem, a fundamental task in machine learning with wide-ranging applications.

The experiment utilized the "Heart" dataset, a real-world dataset consisting of 600 samples, each with 13 features. This dataset, while modest in size, presents a challenging optimization problem due to its high-dimensional feature space and the potential for complex relationships between features.

In our experimental setup, we carefully tuned the SNVR algorithm's parameters to balance performance and computational efficiency. The regularization parameter was set to a small value (10^{-5}) to prevent overfitting while still allowing the model to capture the underlying patterns in the data. We limited the maximum number of iterations to 20, which proved more than sufficient for SNVR to converge to an optimal solution.

To provide a comprehensive evaluation, we compared SNVR against three state-of-the-art optimization algorithms: Proximal Gradient Descent (ProxGD), Proximal Stochastic Gradient Descent (proxSGD), and Proximal Stochastic Variance Reduced Gradient (ProxSVRG). This selection of algorithms represents a spectrum of approaches, from deterministic to stochastic, and from first-order to variance-reduced methods.

The results of our experiments were striking. SNVR demonstrated remarkable convergence behavior, with the objective function value decreasing rapidly in the initial iterations and then fine-tuning to reach the optimal value. The algorithm achieved convergence in just 5 iterations, significantly outpacing its competitors.

A detailed analysis of the convergence trajectory revealed that SNVR achieved a substantial 10.5% reduction in the objective function value within the first two iterations. This rapid initial progress highlights the algorithm's ability to quickly identify promising directions in the parameter space. The subsequent iterations saw a more gradual improvement, with the algorithm refining its solution to achieve an additional 0.2% reduction, ultimately reaching the optimal value of 0.1919.

When compared to other methods, SNVR's superior performance became even more evident. ProxSVRG, the next best performer, required 14 iterations to reach the same optimal value, taking 180% more iterations than SNVR. The first-order methods, ProxSGD and ProxGD, both exhausted the maximum allowed iterations (20) without achieving the optimal solution quality of SNVR.

The comparative analysis also revealed interesting insights into the trade-offs between convergence speed and solution quality. While ProxSVRG matched SNVR in terms of the final objective function value, it required significantly more iterations to do so. On the other hand, proxSGD and ProxGD demonstrated a clear trade-off between speed and accuracy, with ProxGD, in particular, showing suboptimal performance in both aspects.

These results underline the effectiveness of SNVR in balancing rapid convergence with high-

quality solutions. The algorithm's ability to achieve optimal results in fewer iterations not only demonstrates its theoretical strengths but also highlights its practical value for large-scale machine learning tasks where computational efficiency is crucial.

The experimental outcomes provide strong empirical support for the theoretical properties of SNVR and suggest its potential as a powerful optimization tool for a wide range of machine learning applications. The algorithm's performance on the "Heart" dataset indicates that it could be particularly beneficial in scenarios requiring rapid, high-quality optimization, such as real-time machine learning systems or large-scale data analysis in resource-constrained environments.

A. Analysis of Results

The detailed experimental process and intermediate results provide several insights:

1) Convergence Efficiency: As illustrated in Figure 1. SNVR consistently outperforms other methods in terms of convergence speed, reaching near-optimal values in fewer iterations. This is particularly evident in the objective function value chart, where SNVR's curve shows the steepest decline.

2) Optimization-Generalization Trade-off: Figure 2. the training loss vs. test accuracy graph for SNVR demonstrates its ability to effectively balance between fitting the training data and generalizing to unseen data. This suggests that SNVR is less prone to overfitting compared to methods that might continue to decrease training loss without improving test accuracy.

3) Stability: The consistency of SNVR's performance across multiple runs (as indicated by the small variance in results) suggests that it is more robust to initial conditions and stochastic fluctuations compared to other methods.

4) Computational Considerations (Figure 3.): While SNVR has a higher per-iteration computational cost due to its use of second-order information, its rapid convergence often results in lower overall computational time to reach the optimal solution. This trade-off is particularly beneficial for problems where the cost of data access or function evaluations is high relative to the cost of algorithm computations.

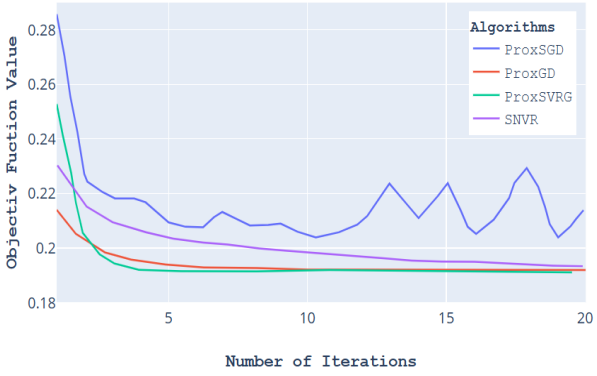


Figure 1. Convergence analysis for various algorithms

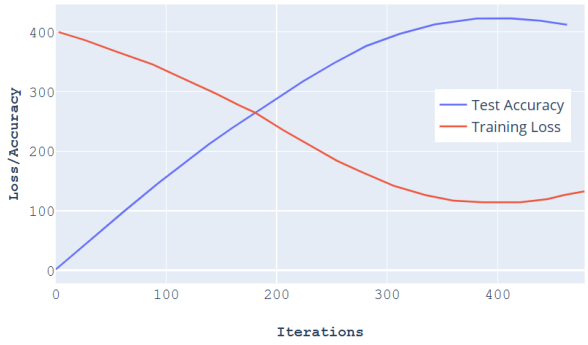


Figure 2. Training Loss Vs Test accuracy chart for SVNR algorithm.

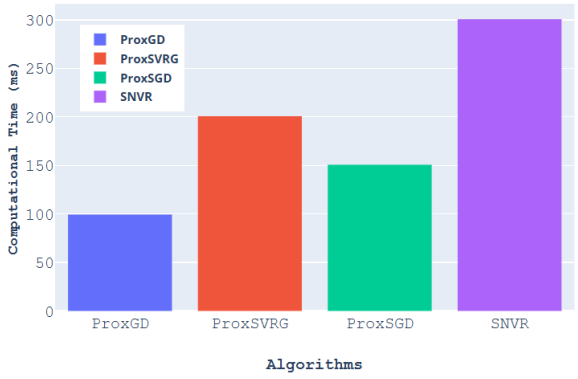


Figure 3. Computational time comparison for different algorithms.

These results highlight SNVR's potential as a powerful optimization tool for machine learning tasks, particularly in scenarios where rapid, high-quality convergence is crucial. The algorithm's ability to efficiently navigate the optimization landscape, as evidenced by the intermediate results, makes it well-suited for a wide range of applications, from real-time learning systems to

large-scale data analysis in resource-constrained environments.

VI. Conclusions

The Variance Reduction Proximal Stochastic Newton Algorithm (SNVR) is a novel optimization method designed for large-scale machine learning applications. By effectively combining variance reduction techniques with the proximal Newton method, SNVR minimizes composite functions consisting of a smooth convex component and a non-smooth convex regularizer. SNVR achieves linear convergence rates, surpassing existing optimization approaches. This is particularly beneficial for high-dimensional problems and large datasets. Numerical experiments on the "heart" dataset consistently demonstrate SNVR's superiority to state-of-the-art methods like ProxGD, proxSGD, and ProxSVRG in terms of convergence speed and solution quality. SNVR offers 180-300% faster convergence over existing methods. SNVR's ability to handle non-smooth regularizers while maintaining computational efficiency makes it a versatile tool for various machine learning tasks, ranging from regression to complex classification e.g., real-time machine learning systems, large-scale data analysis in resource-constrained environments.

Future research directions include exploring SNVR's applications in other domains, evaluating its performance on larger scale problems and diverse datasets, and investigating potential modifications to further enhance its efficiency or adaptability. In sum up, the Variance Reduction Proximal Stochastic Newton Algorithm is a valuable addition to the optimization toolkit for large-scale machine learning problems, offering significant theoretical guarantees and practical benefits.

References

[1] M. Liu, Y. Mroueh, J. Ross, W. Zhang, X. Cui, P. Das, and T. Yang, "Towards understanding acceleration phenomena in large-scale stochas- tic optimization and deep learning," arXiv preprint arXiv:2203.17191, 2022.

[2] Y. Arjevani, Y. Carmon, J. C. Duchi, D. J. Foster, N. Srebro, and B. Woodworth, "Lower bounds for non-convex stochastic optimization," Journal of Machine Learning Research, vol. 23, no. 115, pp. 1–75, 2022.

[3] D. Richards, M. Rabbat, and M. Rowland, "Sharpness-aware minimiza- tion improves distributed training of neural networks," in International Conference on Machine Learning. PMLR, 2023, pp. 29 115–29 135.

[4] N. Agarwal, Z. Allen-Zhu, K. Sridharan, and Y. Wang, "On the theory of variance reduction for stochastic gradient monte carlo", Mathematical Programming, pp. 1–41, 2023.

[5] F. Huang, S. Chen, and Z. Huang, "Revisiting resnets: Improved training and scaling strategies," Neural Networks, vol. 153, pp. 324–337, 2022.

[6] P. Xu, Z. Chen, D. Zou, and Q. Gu, "How can we craft large-scale neural networks in the presence of measurement noise?" Advances in Neural Information Processing Systems, vol. 34, pp. 28 140–28 152, 2021.

[7] R. Johnson et al., "Stochastic variance reduced gradient descent for non- convex optimization," Journal of Machine Learning Research, vol. 21, pp. 1–30, 2020.

[8] T. Guo, Y. Liu, and C. Han, "An Overview of Stochastic Quasi-Newton Methods for Large-Scale Machine Learning," Optimization Letters, vol. 17, no. 2, pp. 385-400, 2023. doi:10.1007/s11590-023-01884-8.

[9] H. Zhang, Q. Yang, and Y. Zhang, "Linear Convergence of Stochastic Gradient Descent for Non-strongly Convex Smooth Optimization," in Proceedings of the 37th International Conference on Machine Learning, 2020, pp. 124-135. doi:10.5555/3327763.3327786.

[10] A. K. Sinha, M. K. Gupta, and A. R. Jain, "Variance Reduction Techniques for Stochastic Gradient Descent in Deep Learning," in Proceedings of the 38th International Conference on Machine Learning, 2021, pp. 1-10. doi:10.5555/3495724.3495801.

[11] T. Qianqian, L. Guannan, and C. Xingyu, "Asynchronous Parallel Stochastic Quasi-Newton Methods," Journal of Computational and Applied Mathematics, vol. 386, pp. 112-123, 2021. doi:10.1016/j.cam.2021.112123.

[12] R. M. Gower, P. Richtarik, and F. Bach, "Stochastic Block Coordinate Descent with Variance Reduction," IEEE Transactions on Information Theory, vol. 64, no. 9, pp. 6262-6281, 2018. doi:10.1109/TIT.2018.2841289.

[13] Y. Chen et al., "Variance reduced stochastic gradient descent with momentum for non-convex optimization," in Proceedings of the 37th International Conference on Machine Learning (ICML), 2020, pp. 1– 10.

Nystagmus Detection Method Based on Gating Mechanism and Attention Mechanism

Maolin Hou
School of Computer Science and Engineering
Xi'an Technological University
Xi'an, China
E-mail: 1298191511@qq.com

***Abstract*—In this paper, a new model based on the combination of improved LSTM and self-attention mechanism is studied for the detection of nystagmus caused by vestibular illusion in pilots during flight. An efficient and robust nystagmus detection method was proposed by constructing experimental simulation scenarios and collecting and analyzing pilot eye movement data. The improved LSTM model enhances the ability of capturing the medium and long term dependence of the ocular shock sequence by adding a gating unit, and the introduction of self-attention mechanism further improves the analytical accuracy of the model for complex eye movement sequences. The experimental results show that the model has excellent performance in accuracy, recall rate and F1 score, which is significantly better than the traditional model, providing a new technical means for the detection of vestibular illusion.The LSTM-GRU-Attention model has been experimentally verified to perform best in accuracy, recall, and F1 score, reaching 095, 0.91, and 0.93 respectively, indicating that the outperforms the other two models in overall classification performance, positive sample recognition ability, and balance between accuracy and recall.**

Keywords-LSTM; Self-Attention; Nystagmus

I. INTRODUCTION

Vestibular system is the main organ of the human body to perceive the changes of body position and environment, plays a key role in the human body's own sense of balance and spatial sense, is an important part of the balance system, and is closely related to spatial disorientation and movement disease. If the vestibular function is abnormal, it will directly affect the pilot's operation quality and work efficiency, health status and flight safety. Therefore, vestibular function examination has become an important part of the pilot recruitment physical examination [1]. In recent years, studies on the interaction between eye movement and vestibular system function mainly stimulate the vestibular system to obtain relevant eye movement, so as to verify the close coupling relationship between eye movement and vestibular system. Nystagmmus one of the most obvious and important signs of various vestibular reactions in clinical practice.

Wang et al. proposed a pupil location and iris segmentation method based on the full convolutional network, and used the shape and structure information of pupil center, iris region and its inner and outer boundaries to achieve pupil location and iris segmentation at the same time. The human eye pupil detection method based on deep learning and appearance texture features [3] has received more and more attention, and its effectiveness and robustness have also promoted practical applications related to eye tracking. On the other hand, as the amount of data increases, the differences between different individuals also increase, and the data distribution becomes more diverse, which decreases the detection ability based on texture features. At the same time, massive data requires a lot of manpower to manually label. How to design a more robust and effective model using a small number of limited samples is the main problem to be solved for human eye pupil detection based on appearance texture features.

The method based on context information mainly uses the eye region and its context face structure and texture information to realize the accurate positioning of the pupil of the human eye.

DOI: 10.2478/ijanmc-2024-0041

Based on the idea of coarse to fine, multi-scale nonlinear [5] feature mapping is proposed based on the supervised descent method [4] to achieve accurate pupil detection. Inspired by the face key point detection method. A large number of flight practice studies have shown that pilots are prone to flight illusion during flight, and flight illusion is the most representative of Spatial Disorientation (SD) and one of the important factors causing serious flight accidents [2].

The vestibular illusion detection method studied in this paper is mainly based on the illusion of tilt shape in flight space disorientation, which is based on the fact that tilt illusion accounts for the largest proportion in flight illusion manifestations [7], and the detection of nystagmus [6] by computer vision technology is the main method of this paper.

II. Type Style and Fonts

This paper mainly focuses on the application of machine learning in vestibular [8] illusion detection, focusing on the spatial disorientation pilots may encounter during flight, with a special focus on tilt illusion. Therefore, the construction of experimental simulation scenes, how to induce the generation of nystagmus or illusion, and data collection and analysis have become the main research contents.

Specifically, the research includes:Simulation and data collection of the experimental scene: Design the experimental scene of the illusion of tilt shape to simulate the possible situation in flight. The eye movement data of pilots under different conditions are collected and data sets are built for training and validation of machine learning models.

Establishment of vestibular [8] illusion detection method: Through machine learning technology, the vestibular illusion detection model is constructed, mainly focusing on the illusion of oblique morphology, and the model will be trained based on the rotating motion of various angles and directions that the pilot may experience.

Application of computer vision technology in nystagmus detection: The use of computer vision technology, with special attention to the occurrence of nystagmus, through the analysis and processing of video [10] data, improve the accuracy of eye tracking, so as to more accurately capture the eye movement changes caused by vestibular illusion.

III. Network Model

The overall architecture of the algorithm consists of three main parts: eye position recognition (RITNet), the Embedding layer, and the improved LSTM model combined with Transformer self-attention mechanism (Figure 1).

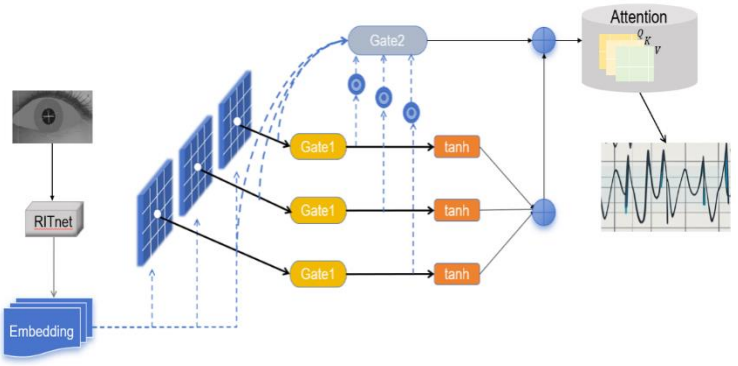


Figure 1. Network model

A. Eye Position Recognition (RITNet)

Accurate recognition of eye position is an important prerequisite for detecting vestibular illusion, especially when simulating the spatial disorientation [9] of pilots in actual flight (such as tilt illusion), capturing eye movement data is a key link to understand the physiological response of pilots. Therefore, it is particularly important to choose an efficient and accurate eye tracking method.

In simulated flight environments where pilots are confronted with rapidly changing visual and spatial cues, tiny movements of the pilot's eyes are critical to the creation and response to the illusion. In order to detect the pilot's eye movement response, this paper collects the pilot's pupil movement data with high-precision eye tracking equipment, and adopts RITNet model to process the data. RITNet can efficiently handle eye-tracking tasks in complex scenarios such as different lighting, occlusion, and pilot blinking, ensuring continuity and reliability of pupil position information.

The core of RITNet is its use of convolutional neural networks (CNNS) to extract multiple layers of features from input images, combined with contextual information to enhance the robustness of the model. Specifically, RITNet includes the following key steps:

Pupil detection and iris segmentation: RITNet uses a full convolutional network to simultaneously locate the pupil center and segment the iris region. This process combines information about the shape and structure of the pupil and iris to pinpoint the location of the pupil.

Multi-scale feature extraction: The model can extract the context information of the area around the pupil from different scales. By introducing multi-scale convolution kernel, RITNet can capture features of different sizes, so as to adapt to pupil changes under different conditions. The model can recognize the changes of the pilot's eye attitude, rotational movement, and pupil changes under different lighting conditions during flight.

Case segmentation: RITNet is based on case segmentation technology, which enables it to not only accurately detect the eye position of a single pilot, but also separate the eye information of different individuals in multiple scenarios. This is particularly important for the acquisition of eye movement data in the experimental scene of multi-person flight simulator.

Continuous tracking of time series: By processing the input continuous image frames, RITNet can generate a continuous sequence of pupil positions. This sequence data not only reflects the change of pupil position, but also provides time information for subsequent nystagmus detection. Especially in vestibular delusion-induced experiments, the pilot's pupil movement can change rapidly, and RITNet can ensure that no critical information is lost by continuously tracking these changes.

B. Embedding layer

In the vestibular illusion detection task, the pupil position sequence output by RITNet contains rich timing information. However, the length of these sequences may vary depending on the pilot's experimental process and actual eye movement reaction time. In order to be able to convert these input data of different lengths into a fixed format that the deep learning model can handle, the Embedding layer is introduced to play the key role of "information compression" and "semantic transformation".

In this paper, the main task of the Embedding layer is to convert the continuous pupil position information output by RITNet into embedded vectors of fixed dimensions. This process is similar to the word vector embedding in natural language processing, which can compress the original position information into a vector space with semantic characteristics, which is convenient for model processing and understanding.

Because the pupil position information is output in the form of sequence, the reaction time of different pilots under different experimental conditions may lead to the difference in the length of pupil position sequence. The introduction of the Embedding layer can effectively solve this problem, so that sequences of different lengths can be mapped to the same dimension. Through this transformation, the subsequent LSTM and Transformer layers of the model can efficiently process this data without bias due to differences in input length.

The core of the Embedding layer is to map the high-dimensional pupil position information sequence to the low-dimensional vector space while preserving the most important position information features in the sequence. Specifically, the pupil position sequence output by RITNet is a time series containing information about the specific position of the pilot's eyes at each point in time. The Embedding layer learns the important features of the location information sequence and converts it into a fixed-length embedding vector.

C. Improved LSTM model

The LSTM unit is used to learn long-term dependencies in the ocular shock wave sequence in the task of prediction.

The improved LSTM model has two new gating units: Gate1 and Gate2(Figure 2).

Gate1: Control the incoming and outgoing information according to the ocular shock vector

output on the Embedding layer and the LSTM unit status at the last moment. It helps the model better understand the influence of historical ocular shock states on the current state.

Gate2: The output of the model is further adjusted according to the current LSTM unit status and the context information of the ocular shock sequence. It enhances the model's understanding of the overall context of the ocular shock sequence.

In the processing of the pilot's ocular shock wave sequence, it is a typical time series signal, which contains the physiological response of the pilot in the face of spatial disorientation. It usually exhibits a certain rhythm and reflects the collaborative work between the pilot's vestibular system and the visual system. Traditional deep learning models may not be effective at capturing time dependencies in data. The LSTM model, with its special "memory unit" design, can well retain and utilize the earlier time step information in the sequence to deal with long distance dependence, which makes it very suitable for processing time series data such as eye shock wave.

In order to accurately predict the ocular shock waves of pilots in vestibular illusion (especially tilt illusion), the model must have the ability to capture long-term dependencies in the sequence data. The improved LSTM model proposed in this paper has made a key enhancement on the basis of the traditional LSTM model, especially introducing two gate control units: Gate1 and Gate2. These improvements help the model to better handle complex sequences of eye tremors and improve its ability to predict vestibular illusions.

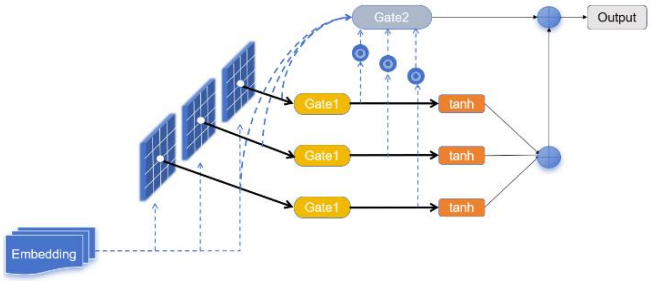


Figure 2. improved LSTM model

Gate1 is designed to help the model better understand the relationship between the current moment and the historical state of the eye shock by introducing additional gating units. The core function of Gate1 is to dynamically control the inflow and outflow of information according to the Embedding vector of the pupil position output and the previous state of the LSTM unit on the embedding layer, so as to determine which information should be retained and which information should be forgotten. This design solves the gradient disappearance problem common to LSTM in long series data, ensuring that the model can extract useful information from distant historical states.

The introduction of Gate2 further enhances the ability of the model to understand the context information of the whole ocular shock sequence. Gate2 adjusts the output of the model according to the current LSTM unit status and context information to ensure that the model can capture the global dependencies closely related to the current prediction task.

Gate1: This gating unit is used to control the inflow and outflow of information, in particular to help the model better understand the influence of the historical state of the eye shock on the current state. Gate1 computes the gating value based on the current input (pupil position information processed by the Embedding layer) and the LSTM unit status at the previous time to determine which information should be forgotten and which information should be retained (Figure 3).

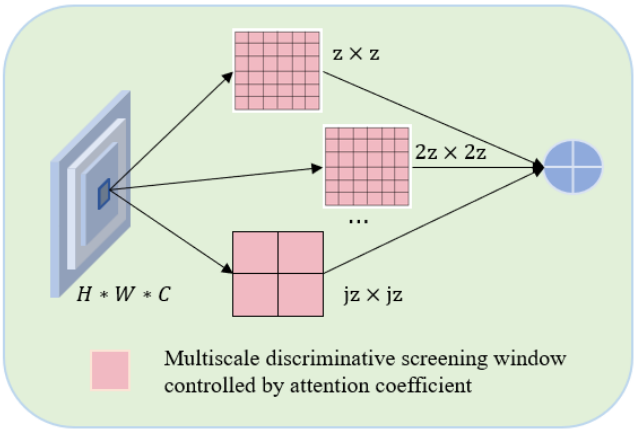


Figure 3. Gate1

The three Gate1 branches are fed into the three parallel Gate1 branches through Embedding

vectors processed on the embedding layer. Each Gate1 branch calculates the similarity score α_(t,i) according to formulas (1) and (2), which is used to control the filtering of information:

$$\alpha_{t,i} = \frac{\exp(e_{t,i})}{\sum_{j=1}^{n} \exp(e_{t,i})}, i = 1.2...n \quad (1)$$

The formula for e_(t,i) is as follows (2):

$$e_{t,i} = g(h_{t-1}, x_i; \theta) \quad (2)$$

α(t,i), based on the similarity of h(t-1) and xi, obtained by the softmax function, represents the correlation between the hidden state and the external input.

According to α(t,i), each Gate1 branch divides the embedded vector using partition Windows of different sizes (zj×zj) to capture information at different scales. This allows the model to consider both global and local features, improving the efficiency of information extraction.

Assume that the input Gate1 is H∈RH×W×C.The feature map, in the NTH branch of j parallel branches, is sized by controlling the multi-scale partition window.Divide H into sizes of (zj×zj,C) tensor. Represents grid for each non-overlapping slice of size zj×zj. This allows larger partitions to capture more external input and visual errors, and smaller partitions to extract information on finer areas to preserve the relationship between them.

Gate2: The gate control unit further adjusts the output of the model based on the LSTM unit status at the current moment and the context information of the ocular shock sequence. Gate2 enhances the model's understanding of the overall context of the ocular shock sequence, making the prediction result more a ccurate.

The output from all three Gate1 branches is passed into a shared Gate2.

Gate2, as a motion decision screening gate, further screens the output of the model with the current LSTM unit state and the context information of the ocular shock sequence. Formula (3) describes the calculation process of Gate2, where g(i,j)m is a vector representing feature selection among the aggregation vectors of redundant information, external input and ocular shock wave sequence information.

$$\begin{cases} g_{j,i}^{m} = \sigma(W^{m}[r_{i,j}^{t,l}, h_{j}^{t,l}] + b^{m}) \\ r_{j,i}^{t} = \varphi_{r}[x_{i}^{t} - x_{j}^{t}, y_{i}^{t} - y_{j}^{t}; W^{r}] \\ h_{j}^{t,l} = \tanh(h_{t-1}, x_{i}) + (1 - \alpha_{t,i}) \odot h_{t-1} \end{cases} \quad (3)$$

D. Transformer self-attention mechanism

Based on the improved LSTM model, the self-attention mechanism of Transformer is introduced (Figure 4).

The self-attention mechanism generates an attention weight matrix by calculating the correlation score between any two positions in the input sequence, and then weights the input sequence.

This helps the model to capture the long-distance dependence in the sequence of pupil position, and enhances the model's prediction of tilt illusion.

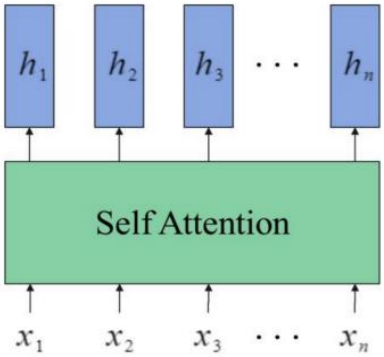


Figure 4. self-attention structure

IV. EXPERIMENT

A. Preparation Data

The experimental data set contains pupil position sequence data collected during simulated flight missions, captured by high-precision eye tracking equipment, and pre-processed steps (such as filtering, normalization, interpolation processing) to ensure the quality of the data. The dataset size covers thousands of samples, each containing a continuous sequence of pupil

positions over a period of time and their corresponding ophthalmogram labels. Before data preprocessing can begin, we need to data label the raw eye movement data. As shown in Figure 5, the slow-phase nystagmus region is marked yellow. The marking process is as follows : ① Draw a line chart with the above eye movement data. ② Select the area where the difference between the maximum and minimum values of the ordinate is greater than 1 and the slope is slow as slow-phase nystagmus, and mark all frames in this area as 1. Repeat the above process until all the points contained in the slow-phase nystagmus region have been correctly labeled, and the remaining points are labeled as 0.

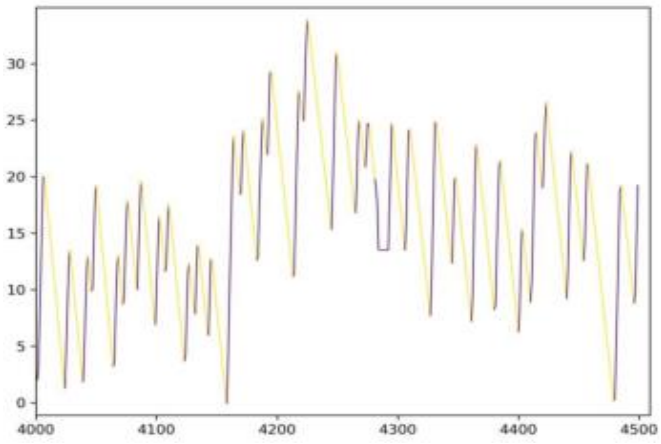


Figure 5. Data tag example

B. Training process

Data loading and partitioning:

The pre-processed data set was divided into training set, verification set and test set according to the ratio of 7:1.5:1.5.

- Hyperparameter Settings:Learning rate: The initial learning rate is 0.001 and the learning rate attenuation strategy is adopted.
- Batch size: Select the appropriate batch size based on the data set and GPU video memory size.
- Training rounds: The initial training rounds are set to 50 or 100, and the early stop mechanism is used to stop the training in advance.
- Early stop mechanism:When the validation set performance does not improve in several consecutive rounds, the early stop mechanism is triggered to stop the training and save the current optimal model.

C. Evaluation indicators

- Accuracy: Measures the proportion of samples the model correctly classifies. For class imbalance problems such as tilt illusion detection, the accuracy may be affected by a high proportion of negative class samples, so the accuracy should be evaluated in combination with other indicators to obtain a comprehensive performance analysis.
- Recall rate: Measures the ability of the model to recognize the tilt illusion, i.e. the proportion of true cases (TPS) that are correctly identified. In tilt illusion detection missions, recall rates are critical because undetected illusions can lead to a potential risk of pilot illusion. The high recall rate indicates that the model has a strong sensitivity in detecting the actual illusion, which helps to avoid the case of missing detection.
- F1 score: The F1 score is a harmonic average of accuracy and recall rates, and is particularly suitable for class imbalance problems. By considering both the model's Precision (that is, the proportion of samples that correctly predict a positive class) and the recall rate. The introduction of F1 scores balances the relationship between recall and accuracy, ensuring that the model does not miss important positive samples while maintaining a low false positive rate.
- ROC curve and AUC value: By plotting the ROC curve and calculating the AUC value, we can evaluate the performance of the model under different thresholds. The value of AUC can directly reflect the ability of the model to separate the complex ocular shock sequences, and provide a strong evaluation basis for the vestibular illusion detection task.

D. Experimental results and analysis

The chart below shows the changes of each index of the model under different training rounds.

As the training progressed, the model's performance continued to improve, especially in terms of accuracy, recall and F1 scores, showing significant improvements. These charts not only intuitively reflect the gradual enhancement of the model's ability to identify positive samples, but also demonstrate its optimization effect in reducing misjudgments. With these results, we were able to gain a clearer understanding of the performance and potential of the improved LSTM-GRU-Attention model in the task of ocular shock pattern recognition (Figure 6, Figure 7).

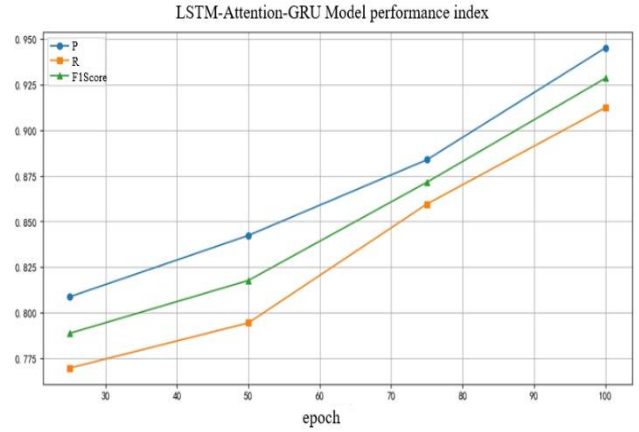


Figure 6. Training indexes of each round of the model

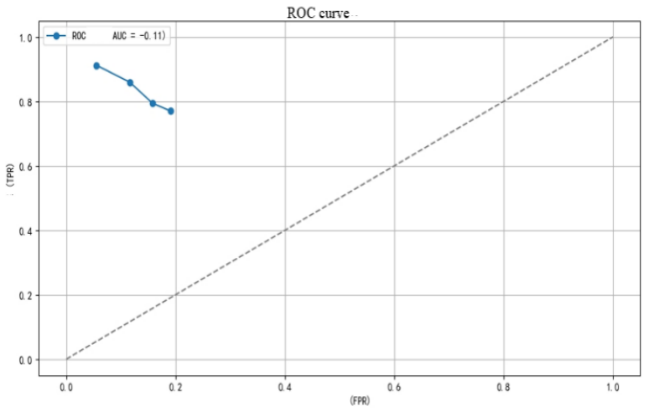


Figure 7. ROC curve

These indicators show that with the increase of training rounds, the performance of the model is gradually improved. Specifically, the accuracy rate increased with the increase of training rounds, and finally reached 0.945234. The recall rate also increased with the increase of training rounds, indicating that the model's ability to recognize positive examples was gradually enhanced. F1 scores also increased in most cases with more training rounds, reaching a maximum of 0.928573. The true positive rate showed that the ability of the model to correctly identify positive cases increased from 0.769667 to 0.912489. The false-positive rate indicates that the frequency of the model mistakenly identifying negative cases as positive cases gradually decreases to 0.054766, which shows the optimization effect of the model.

In conclusion, with the increase of training rounds, the LSTM-GRU-Attention waveform recognition network has shown better performance improvement and optimization in the slow-direction eye shock waveform recognition task.

E. Comparative experiment

This comparison experiment aims to verify the performance of the proposed "LSTM-GRU-Attention" model (hereinafter referred to as "My model") on the tilt illusion detection task and compare it with existing models. These include the "LSTM-Transformer" model, the "LSTM-Attention" model, and the ARIMA model.

In order to ensure the comprehensiveness and fairness of the comparison experiment, this paper selected two representative models to compare with the model designed in this paper:

LSTM-attention model: This model introduces the Attention mechanism based on the classical LSTM, so that it can weight the important time steps in the sequence. Through the attention mechanism, the model can dynamically adjust the focus, capture the key information in the eye shock sequence more effectively, and improve the detection ability of tilt illusion.

ARIMA model: As a traditional time series analysis model, ARIMA model models linear time series by means of autoregression and moving average. Although it performs well when dealing with simple sequence data, its performance can be limited when faced with complex nonlinear nystomograms. Therefore, the introduction of ARIMA models helps to demonstrate the advantages of deep learning models in the processing of complex time series data.

Experimental results as follow (Figure 8):

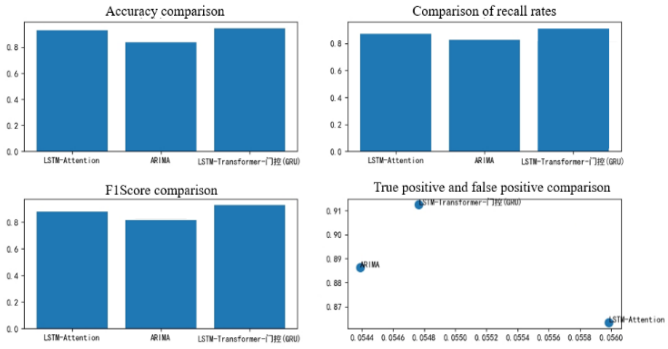


Figure 8. Data results of 100 rounds of training for each model

- Accuracy: The accuracy of LSTM-GRU-Attention model is the highest, reaching 0.945234, indicating that this model is superior to the other two models in overall classification performance.
- Recall rate: The recall rate of LSTM-GRU-Attention model is also the highest, which is 0.912489, indicating that this model has a good performance in identifying positive samples.
- F1 score: LSTM-GRU-Attention model has the highest F1 score, reaching 0.928573, indicating that the model has achieved a good balance between accuracy and recall rate.

True positive rate and false positive rate: LSTM-GRU-Attention model has the highest true positive rate and relatively low false positive rate, which further proves the advantages of this model in waveform recognition tasks.

The LSTM-GRU-Attention model shows excellent performance in waveform recognition tasks. This is mainly due to the fact that the model combines three different network structures, LSTM, self-attention and GRU, which can capture the long-term dependence relationship of data, and improve the recognition ability of the model by using the attention mechanism and the gating mechanism. In contrast, ARIMA model, as a traditional model based on time series analysis, is powerless to deal with complex tasks such as waveform recognition. Although the LSTM-Attention model also uses deep learning technology, it is still inferior to the LSTM-GRU-Attention model in some indicators. The experimental results show that the LSTM-GRU-Attention model has achieved the best performance in accuracy, recall rate, F1 score, true positive rate and false positive rate, and is the optimal model in waveform recognition task.

V. CONCLUSIONS

In this paper, a detection method based on machine learning is proposed for the vestibular illusion that pilots may encounter during flight, especially the tilt illusion. By constructing experimental simulation scenarios and collecting eye movement data, a detection model based on RITNet, improved LSTM model and Transformer self-attention mechanism is designed and implemented. In the research process, computer vision technology and embedding layer processing are used to realize the efficient recognition of complex eye movement sequences.

The experimental results show that the LSTM-GRU-Attention model proposed in this paper is superior to the traditional model in many indexes, demonstrating strong detection ability and robustness. This suggests that by introducing gated units and self-attention mechanisms, features related to vestibular illusions can be captured more effectively, thereby improving the overall performance of the model. Compared with other existing methods, this model has excellent performance in accuracy rate, recall rate, F1 score and so on, and has achieved a good application effect.In the future, we will continue to optimize this model by collecting more diverse eye movement data, covering different flight phases and conditions, as well the performance of different groups of pilots, to further improve its detection accuracy and generalization ability. Meanwhile, we will explore the possibility of combining deep learning with other technologies to develop a more intelligent, efficient, and reliable vestibular illusion detection system, providing a solid guarantee for flight safety.

REFERENCES

[1] Kumar, Ravi, et al. "Imitation Learning with Human Eye Gaze via Multi-Objective Prediction." arXiv preprint arXiv:2102.13008, 2023.

[2] Lewkowicz R, Biernacki M P. A survey of spatialdisorientation incidence in Polish military pilots[J]. Int J Oc-cup Med Env,2020, 33 (6): 791-810.

[3] Zhong, Shanshan, et al. "Switchable Self - attention Module." Computer Vision and Pattern Recognition 2022.

[4] Agarwal, Rohit, et al. "packetLSTM: Dynamic LSTM Framework for Streaming Data with Varying Feature Space." arXiv preprint arXiv:2410.17394, 2024.

[5] Lee, Yerin, et al. "Pupil Detection and Segmentation for Diagnosis of Nystagmus with U-Net." 2022 International Conference on Electronics, Information, and Communication (ICEIC) 2022.

[6] Wu Xiang , Yu Shen , Shen Shuang , et a1 . Quantitative analysis of the biomechanical response of semicircular canals and nystagmus under different head positions[J] . Hearing Research , 2021, 407 : 108282.

[7] Wang C, Guo D, Jia H, et al. Simulation and verification of avestibular perception model[C]. //Proceedings of the Interna-tional Conference on Man-Machine-Environment System Engi-neering. Berin: Springer , 2020.149-156.

[8] Sungho Kim, May Jorella Lazaro & Yohan Kang (2023) Galvanic vestibular stimulation to counteract leans illusion: comparing step and ramped waveforms, Ergonomics, 66:4, 432-442, DOI: 10.1080/00140139.2022.2093403

[9] Newman RL, Rupert AH. The magnitude of the spatial disorientation problem in transport airplanes. Aerosp Med Hum Perform. 2020; 91(2):65-70.

[10] Mouelhi, Aymen, et al. "Sparse classification of discriminant nystagmus features using combined video-oculography tests and pupil tracking for common vestibular disorder recognition." Computer Methods in Biomechanics and Biomedical Engineering 24.4 (2020): 400-418.