Journal of Translational Medicine

RESEARCH

Open Access

SVEA: an accurate model for structural variation detection using multi-channel image encoding and enhanced AlexNet architecture

Taixing Qiu[1,2], Jiawei Li[2], Yan Guo[3], Limin Jiang[3*] and Jijun Tang[2*]

Abstract

Background Structural variations (SVs) are a pervasive and impactful class of genetic variation within the genome, significantly influencing gene function, impacting human health, and contributing to disease. Recent advances in deep learning have shown promise for SV detection; however, current methods still encounter key challenges in effective feature extraction and accurately predicting complex variations.

Methods We introduce SVEA, an advanced deep learning model designed to address these challenges. SVEA employs a novel multi-channel image encoding approach that transforms SVs into multi-dimensional image formats, improving the model's ability to capture subtle genomic variations. Additionally, SVEA integrates multi-head self-attention mechanisms and multi-scale convolution modules, enhancing its ability to capture global context and multi-scale features. The model was trained and tested on a diverse range of genomic datasets to evaluate its accuracy and generalizability.

Results SVEA demonstrated superior performance in detecting complex SVs compared to existing methods, with improved accuracy across various genomic regions. The multi-channel encoding and advanced feature extraction techniques contributed to the model's enhanced ability to predict subtle and complex variations.

Conclusions This study presents SVEA, a deep learning model incorporating advanced encoding and feature extraction techniques to enhance structural variation prediction. The model demonstrates high accuracy, outperforming existing methods by approximately 4%, while also identifying areas for further optimization.

Keywords Structural variations, Deep learning, Multi-head Self-attention mechanism, Multi-channel encoding

Introduction

Structural variations (SVs) are a major form of genetic variation, typically involving more than 50 base pairs (bps) in structural changes, such as deletions (DELs), insertions (INSs), duplications (DUPs), and inversions (INVs) [1]. Along with single nucleotide polymorphisms (SNPs) and small insertions/deletions (Indels), SVs are a major component of human genomic diversity. However, due to the larger number of base pairs involved, their impact on gene function and human health is often more significant [2]. For instance, repeat sequence expansion can lead to gene overexpression, causing abnormal protein accumulation, which may

*Correspondence:
Limin Jiang
lxj423@med.miami.edu
Jijun Tang
jj.tang@siat.ac.cn
[1] College of Engineering, Southern University of Science and Technology, Shenzhen 518055, China
[2] Faculty of Computer Science and Control Engineering, Shenzhen University of Advanced Technology, Shenzhen 518055, China
[3] Department of Public Health Sciences, University of Miami, Miami, FL 33136, USA

result in diseases such as Parkinson's disease [3] and Alzheimer's disease [4]. Furthermore, gene deletions affecting the expression of essential genes can lead to abnormalities in brain structure and function, closely associated with neurodevelopmental disorders such as intellectual disability [5], autism, and schizophrenia [6]. Chromosomal inversions can disrupt the normal structure of genes, leading to abnormal expression and causing diseases such as hemophilia A [7].

Currently, computational methods of structural variation (SV) detection are primarily divided into three categories: alignment-based, assembly-based, and deep learning-based approaches.

Alignment-based methods directly extract SV features from alignment information and form consensus SV calls by combining overlapping reads. The primary advantage of these methods lies in their computational efficiency, particularly with large-scale genomic data, often outperforming assembly-based methods in processing speed. Tools such as cuteSV[8], Sniffles [9], SVIM [10], and DeBreak [11] utilize alignment information generated by aligners to rapidly call SVs, making them well-suited for low-coverage datasets,On the HG002 Pacbio CLR dataset with a coverage of 69x, the accuracy of cuteSV, Sniffles, and SVIM reached 94.78%, 93.68%, and 93.14%, respectively. On the HG002 ONT dataset with a coverage of 47x, their accuracies were 92.14%, 84.63%, and 85.95%, respectively. On the HG002 Pacbio CCS dataset with a coverage of 28x, the accuracies were 94.59%, 93.65%, and 88.89%, respectively. In contrast, on the NA19240 dataset with a coverage of 40x, the accuracies dropped to 65.62%, 60.23%, and 56.53%, respectively. Additionally, Debreak achieved accuracies of 93.36% for INS and 96.48% for DEL on the HG002 Pacbio CLR dataset. Additionally, the workflow for alignment-based methods is relatively mature, leveraging existing alignment tools like Minimap2 [12] and NGMLR [9]. However, these methods heavily rely on the accuracy of the reference genome, making them less effective for detecting complex SVs, such as large insertions and inversions. Moreover, the choice of alignment algorithm and parameter settings can significantly impact the results.Alignment-based methods have certain limitations. First, these methods heavily rely on the accuracy of the reference genome, and when the reference genome is incomplete, it can affect the detection of variants. Secondly, these methods face difficulties in detecting complex structural variations, especially when these variations result in ambiguous alignment information. Additionally, low-coverage data can also impact their performance, as insufficient alignment information may lead to incorrect SV calls.

Assembly-based methods generate consensus sequences either de novo or through reference-guided assembly and then extract SV features by aligning the consensus sequences to the reference genome. These methods can identify large-scale genomic alterations and complex variants, especially those that are difficult for alignment-based algorithms to detect [13]. Assembly-based methods perform particularly well in high-repetitive regions and maintain high accuracy even in the presence of reference genome biases. However, assembly processes are computationally intensive and perform poorly with low-coverage data. Additionally, the runtime of assembly tools can be significant, particularly when deep sequencing data is required, resulting in high computational costs [14].

Deep learning-based methods have made significant progress in detecting both small variants and SVs in recent years. These methods use model inference to classify variants, surpassing traditional rule-based approaches. For example, BreakNet [15] and MAMnet [16] combine convolutional neural networks (CNNs) with long short-term memory (LSTM) networks to extract features from small regions of the genome and predict variants. SVision [17] converts alignment information into images and uses CNNs to predict the probability of SVs, while Cue [18] processes short-read data by juxtaposing genomic intervals to generate input images and employs a stacked hourglass network to predict breakpoint locations.

Despite these advances, challenges remain for deep learning methods. For example, certain encoding methods are optimized for specific variant types, potentially overlooking hidden features in the alignment region. For instance, models like DeepVariant [19], which focus on detecting small variants (such as SNPs and Indels), perform well in encoding these types of variants, but their encoding methods fail to capture the features of larger structural variants , resulting in inaccurate detection of complex variants. Furthermore, models like SVNet [20] use relatively shallow architectures with fewer convolutional layers, making it difficult for them to capture long-range dependencies in genomic data. As a result, while they may perform well in detecting small variants, their ability to detect more complex variants is limited, especially when large-range dependencies exist in the genome. Additionally, some models, such as BreakNet, need to model long-range upstream and downstream information in genomic data to predict the precise boundaries of structural variants. Shallow networks, however, struggle to effectively capture these dependencies, causing the model to fail in accurately predicting the impact area of structural variants, especially in cases with large variation regions.

Against this backdrop, we developed a deep learning-based SV prediction model called SVEA (Structural Variation detection with Enhanced AlexNet drchitecture). SVEA leverages alignment information from Concise Idiosyncratic Gapped Alignment Report (CIGAR) [21] strings and employs an enhanced multi-channel image encoding method to fully utilize all available information in the target region, thereby improving the model's accuracy and generalization ability for detecting complex variants across different genomic regions. Built on AlexNet, the SVEA model incorporates a multi-head self-attention mechanism (MHSA) [22] and multi-scale convolutional modules [23] to enhance its ability to capture global context and multi-scale features.

Materials and methods

The workflow of SVEA, as shown in Fig. 1, consists of three main stages: (1) Data preprocessing: Extracting target region information from long-read alignment files in BAM format. (2) Embedding technology for alignment results: Converting the extracted data into three-channel images using a specific encoding strategy. (3) Prediction model: Training an enhanced AlexNet model incorporating a multi-head attention mechanism. After training, the model is applied to new genomic data to predict and identify the type and location of structural variations, ultimately generating the prediction results.

Data preprocessing

Existing methods for detecting structural variations often rely on specific encoding strategies tailored to different types of variations. This reliance on predefined filtering approaches may overlook latent feature information present within the alignment regions. To mitigate this limitation, we propose an optimized encoding method that directly utilizes alignment information, thereby reducing dependency on predefined variation types. This method enables the model to autonomously

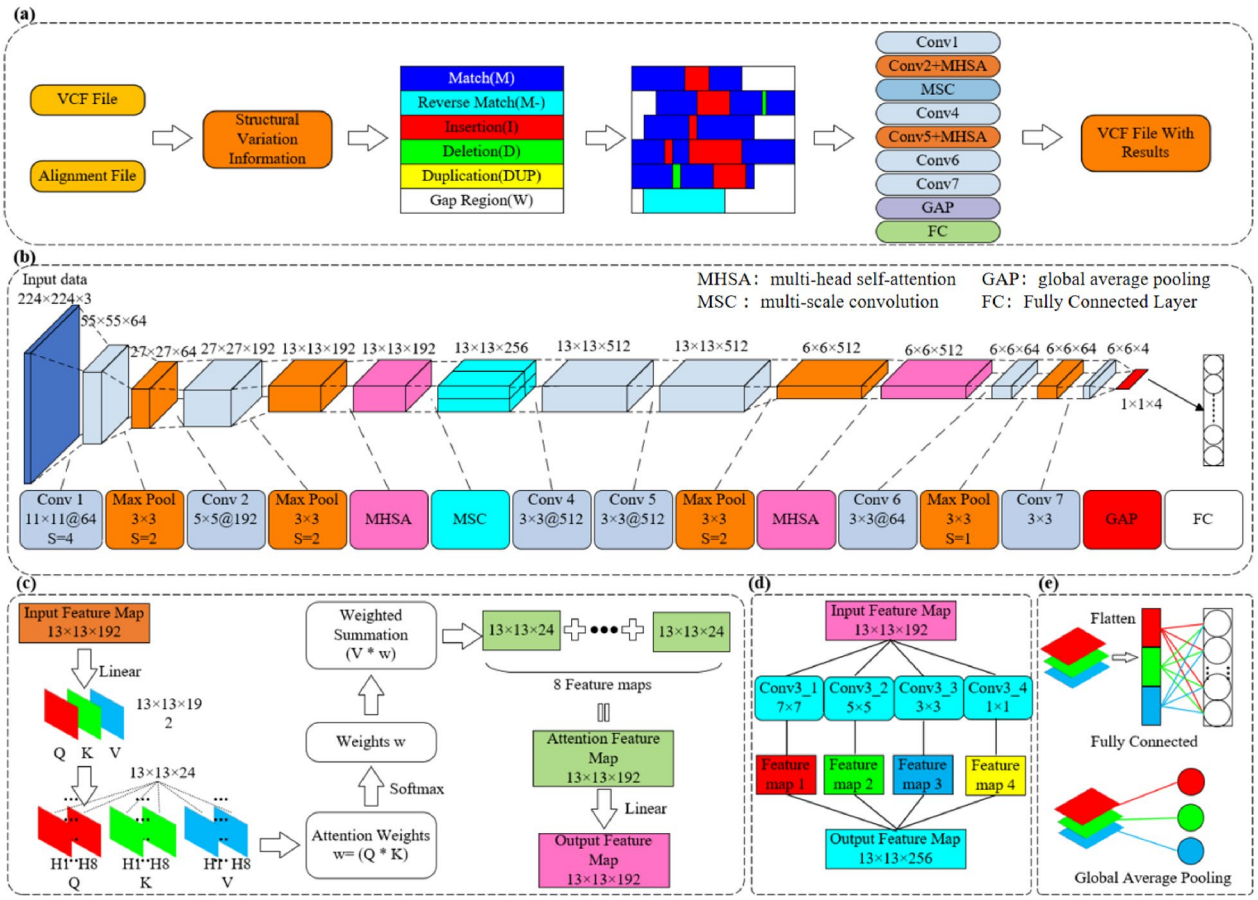

Fig. 1 Workflow of SVEA and the architecture of the Enhanced AlexNet model. **a** Workflow of SVEA. **b** Detailed parameters of the Enhanced AlexNet model. **c** Principle of the multi-head self-attention mechanism. **d** Operational principle of the multi-scale convolutional module. **e** Comparison between the global average pooling layer and the fully connected layer. MHSA is multi-head self-attention; MSC is multi-scale convolution; GAP is global average pooling; FC is fully connected layer

learn and extract structural variation features from the alignment data, fully leveraging all available information in the target regions and enhancing the model's accuracy and generalization capability for complex variations across diverse genomic regions. We will outline the specific implementation steps of this encoding method in detail.

To comprehensively extract structural variation information from VCF files, we first generate high-confidence VCF files using SV callers or obtain rigorously validated truth set VCF files from relevant databases. Next, we design and implement an automated parsing script to systematically extract key feature information from the VCF files, including chromosome name (chrname), SV position (SV pos), variation type (SVTYPE), variation length (SVLEN), and end position (SVEND). All extracted data are stored in a dictionary data structure, with each SV represented as an individual dictionary object. These dictionaries are then compiled into a list for further in-depth analysis and research.

We dynamically adjust the read regions by setting their length to three times that of the structural variation. The selection is centered on the structural variation position. Given that, in VCF files, the end position for INS-type structural variations is typically the same as the SV pos, we extend one SV length beyond SV pos to determine the end position of the read segment (SV end). Next, we extend one SV length backward from the SV pos and another SV length forward from the SV end to form the final read region (sv left, sv right). We then retain the read segments that include these regions for analysis, as illustrated in Fig. 2a.

Embedding technology for alignment results

We extract read segments from the BAM file based on the left and right boundaries (sv left, sv right) of the read region and parse the CIGAR string of each segment, focusing on "match" (M), "insertion" (I), and "deletion" (D) operations. First, we identify the insertion positions and deletion intervals based on the segment's coordinates and record the normally matched segments. For insertion operations, we log their positions and lengths in the reference sequence and separately process the matching segments before and after the insertion. For deletion operations, we record the start and end positions of the deletion and adjust the ranges of the normal matching segments accordingly. Next, we filter the insertion and deletion segments, retaining only those within the defined region and trimming any segments that extend beyond the boundaries. Finally, we adjust the normal

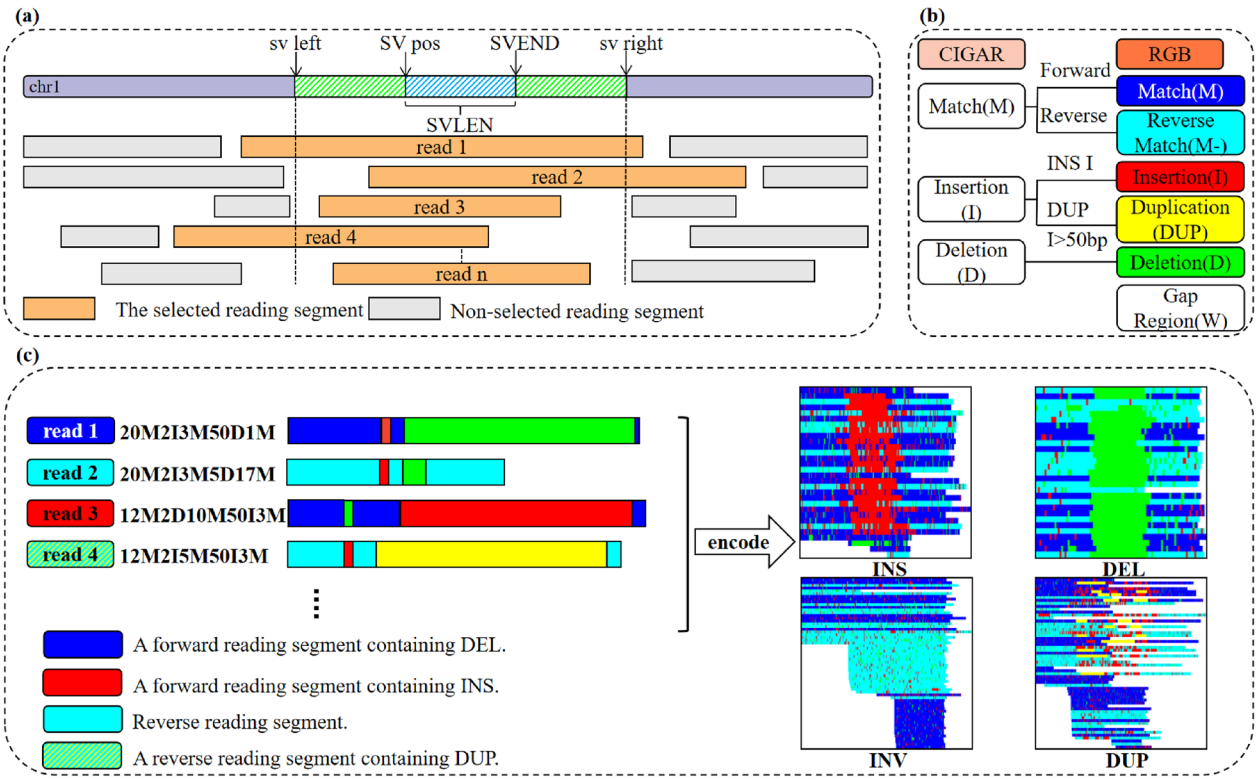

Fig. 2 Encoding process of long-read alignment information. **a** Selection of read regions and read segments. **b** Encoding of CIGAR strings into RGB images. **c** Examples of read segment encoding and instances of the four encoding types

matching segments according to their orientation, marking them as either forward (M+) or reverse match (M-) segments.

For SVTYPE labeled as "DUP", it is difficult to distinguish between insertion (INS) and duplication (DUP) using the CIGAR string alone, because both types of structural variations may be represented similarly in the CIGAR string, such as through insertions or matches . The CIGAR string does not provide enough information about the origin or the exact position of the duplicated sequence, so we mark any insertion segments longer than 50 bps within the SV pos and SV end regions as "DUP". After sorting all segments, we ensure that both boundaries are covered and fill in blank segments (W) where necessary. Finally, as illustrated in Fig. 2b, we generate images by assigning colors to different segment types: red (RGB: 255, 0, 0) for insertions, green (RGB: 0, 255, 0) for deletions, blue (RGB: 0, 0, 255) for normal matches, yellow (RGB: 255, 255, 0) for duplications, cyan (RGB: 0, 255, 255) for reverse matches, and white (RGB: 255, 255, 255) for gap regions.

The width of the image is dynamically adjusted based on the total length of the segments, and the height of each read segment is dynamically adjusted based on the total number of segments, resulting in a 224×224 pixel image for subsequent analysis and processing. The actual images generated for the four types of structural variations are shown in Fig. 2c. For INS-type structural variations, the central regions of multiple segments are encoded in red. Similarly, DEL-type structural variations are encoded in green in the central regions of multiple segments. For INV-type structural variations, most segments are encoded in cyan, as these segments match the reverse strand. In the case of DUP-type structural variations, the central insertion regions of multiple segments are encoded in yellow.

Prediction model based on attention mechanism and convolution network

As shown in Fig. 1b, the enhanced AlexNet model exhibits significant differences from the traditional AlexNet model in several key design aspects.

Firstly, multi-head self-attention mechanism [24] is applied in the second and fifth layers, enabling the model to capture global dependencies between distant features. The traditional AlexNet model primarily relies on local convolution operations, making it less capable of handling long-range feature dependencies. By incorporating multi-head self-attention, the enhanced model significantly improves its ability to process global features and enhance its overall flexibility.

Secondly, multi-scale convolution module [25] is used to significantly enhance the diversity and adaptability of feature extraction. The traditional AlexNet model utilizes convolutional filters of a single size, limiting its capacity to handle features across different scales. In contrast, the enhanced model employs parallel multi-scale convolutions (7×7, 5×5, 3×3, and 1×1), allowing it to extract information from multiple scales and thus more effectively process input images with complex structures.

Thirdly, global average pooling [26] is another notable improvement. In the traditional AlexNet model, fully connected layers are used after the convolutional layers, which often results in a large number of parameters and increases the risk of overfitting, especially when handling large-scale datasets. The enhanced model addresses this by introducing a global average pooling layer at the end, reducing each feature map to a 1×1 dimension. This significantly reduces the parameter count while retaining global feature information, which ultimately lowering the model's complexity and improving generalization.

Finally, the batch normalization [27] is applied in each convolutional layer, which accelerates the training convergence and improves stability. In contrast, the traditional AlexNet model only uses local response normalization (LRN) in certain layers, limiting the overall benefits of normalization techniques.

Through these four key modifications, the enhanced AlexNet model achieves substantial improvements in feature extraction, global information processing, and parameter efficiency compared to the traditional model, particularly excelling in handling complex visual tasks.

Multi-head self-attention mechanism

As shown in Fig. 1c, the multi-head self-attention mechanism processes input features through multiple parallel attention heads. Each attention head independently computes the relationships between different regions by transforming the input features into queries (Q), keys (K), and values (V). The heads individually calculate the attention scores and subsequently apply the softmax function to convert these scores into normalized weights, which are then used to generate the weighted context representation. The specific formula is as follows:

$$\text{Attention}(Q, K, V) = \text{softmax}\left(\frac{QK^{\top}}{\sqrt{d_k}}\right)V \tag{1}$$

The matrices $Q = XW_Q, K = XW_K$, and $V = XW_V$ represent the query, key, and value matrices, respectively, which are generated from the input features X using the weight matrices W_Q, W_K, W_V. The softmax function ensures that the attention scores for each head are normalized into weights, which are then used to compute the weighted sum, producing the final context representation. Here, d_k represents the dimensionality of

the key vectors, and it is used to scale the dot product of the query and key matrices to prevent large values in the softmax function.

The advantage of this multi-head attention mechanism lies in the fact that each attention head can focus on different feature subspaces, capturing relationships across various scales and regions. Through residual connections, the input features are combined with the features processed by the self-attention mechanism, as described by the following equation:

$$\text{Output} = \text{LayerNorm}(X + \text{MultiHead}(Q, K, V)) \tag{2}$$

This residual connection not only preserves the original information but also enhances the expression of global features, improving the model's performance and stability.

In traditional convolution operations, the receptive field is limited by the local convolution kernel, and can only capture relationships between neighboring pixels. The calculation formula is:

$$\text{Conv}(X) = W * X \tag{3}$$

In this case, W represents the convolution kernel, and $*$ denotes the convolution operation. However, the convolution operation is limited by its local receptive field, making it difficult to capture dependencies between distant pixels in the image. In structural anomaly detection tasks, images often contain complex and scattered features, with different anomaly patterns spanning large spatial regions. Relying solely on local information for detection may result in missed detections or false positives, especially when handling dispersed but crucial features.

The multi-head self-attention mechanism allows the model to process information from different feature subspaces in parallel, thereby improving its ability to capture global features. The calculation formula is:

$$\text{MultiHead}(Q, K, V) = \text{Concat}(\text{head}_1, \ldots, \text{head}_h) W_O \tag{4}$$

where the outputs of multiple attention heads (head_i) are concatenated to form the final output representation. This enables the model to effectively combine information from different scales, particularly in structural anomaly detection tasks, where it can capture key features dispersed across various regions of the image. Through this mechanism, multi-head self-attention significantly enhances the model's ability to identify and classify these dispersed features.

Overall, the multi-head self-attention mechanism greatly improves the model's performance in structural anomaly detection. By capturing global dependencies through multiple parallel attention heads and preserving local features via residual connections, this mechanism reduces the model's reliance on local features, mitigates noise interference, and enhances robustness and detection accuracy through the integration of global information.

Multi-scale convolutional module

As shown in Fig. 1d, in structural anomaly detection, different anomaly patterns may be distributed at various scales in the image-some involving large regions, while others are localized or minor changes. By using a multi-scale convolution module, the model can simultaneously extract both large-scale and small-scale image features, ensuring that both broad and local information is captured.

The core design of multi-scale convolution involves using convolution kernels of different sizes in parallel (e.g. 7×7, 5×5, and 3×3) to capture features at different scales. The calculation for the multi-scale convolution module is as follows:

$$F_{\text{multi-scale}} = \text{Concat}(F_{7\times7}, F_{5\times5}, F_{3\times3}) \tag{5}$$

Here, $F_{7\times7}$, $F_{5\times5}$, and $F_{3\times3}$ represent features extracted by different convolution kernels, and these features are concatenated along the channel dimension to form a richer feature representation.

The benefits of this design include: Firstly, the multi-scale convolution module captures both global and local features, allowing the model to detect structural anomalies spread across regions of different sizes in the same layer, facilitating comprehensive evaluation of multiple anomaly types in an image. Second, it enhances the ability to detect complex anomalies, simultaneously addressing large structural changes and minor variations affecting only a few pixels. The convolution operation is calculated as follows:

$$F_{\text{conv}} = W * X \tag{6}$$

where W represents the convolution kernel, X is the input feature, and $*$ denotes the convolution operation.

Additionally, multi-scale convolution increases the model's robustness by avoiding the omission of key information due to the limitations of a single-scale convolution kernel, ensuring the effective detection of anomalies of various shapes and sizes. The 1×1 convolution kernel is used for feature fusion, with the calculation as:

$$F_{1\times1 fusion} = W_{1\times1} * F_{\text{multi-scale}} \tag{7}$$

This convolution kernel serves to fuse features across different scales, helping the model integrate global and local

information extracted from the 7×7, 5×5, and 3×3 kernels, resulting in richer feature representations.

The core of the convolution operation lies in using convolution kernels of different shapes to extract features, and enhancing the model's non-linear expressiveness through appropriate activation functions. One commonly used activation function is ReLU (Rectified Linear Unit), which is defined as:

$$\mathrm{ReLU}(x) = \max(0, x) \tag{8}$$

where x is the input value to the activation function, which can be any real number. The expression $\max(0, x)$ represents the output of the ReLU function, where if x is greater than 0, it outputs x, and if x is less than or equal to 0, it outputs 0.

The ReLU activation function effectively avoids the vanishing gradient problem and has the advantage of high computational efficiency. By introducing a non-linear operation, it enables the network to learn more complex feature representations. In the multi-scale convolution module, the ReLU activation function is applied to the feature maps produced by each convolution kernel, enhancing the network's ability to represent features at different scales.

Through these designs, the multi-scale convolution module significantly enhances the model's performance in structural anomaly detection tasks.

Global average pooling layer

As shown in Fig. 1e, the global average pooling (GAP) layer in convolutional neural networks is used to reduce the spatial dimensions of each feature channel to a single value. Unlike traditional fully connected layers, GAP computes the average of all elements within each channel, significantly reducing the number of parameters. The specific calculation formula is as follows:

$$F_{\mathrm{GAP}} = \frac{1}{H \times W} \sum_{i=1}^{H} \sum_{j=1}^{W} F(i, j) \tag{9}$$

where $F(i, j)$ represents the value at position (i, j) of the feature map, and H and W are the height and width of the feature map, respectively. By averaging the values within each channel, GAP generates a feature vector, avoiding the large number of parameters associated with fully connected layers while retaining global information.

Unlike fully connected layers, GAP does not require learned weights, thus significantly reducing the number of parameters and mitigating the risk of overfitting. The parameter count for fully connected layers is:

$$\mathrm{Params}_{\mathrm{FC}} = C_{\mathrm{in}} \times C_{\mathrm{out}} \times H \times W \tag{10}$$

Here, C_{in} denotes the number of input channels, and C_{out} denotes the number of output channels.

In contrast, GAP has zero parameters, making the model more lightweight. Furthermore, GAP improves the model's generalization ability, particularly in cases of imbalanced datasets or limited data. This pooling technique effectively preserves global information related to structural anomalies, ensuring that the model captures important global features while reducing the parameter count.

Therefore, the GAP layer not only plays a significant role in enhancing model performance and preventing overfitting but also ensures that global features are retained, making it highly effective in handling complex structural anomaly detection tasks.

Batch normalization

In structural anomaly detection tasks, the variation patterns in images may spread across multiple scales and regions. Batch Normalization (BN) normalizes all spatial positions within each feature channel, making the model more robust in handling these cross-scale variations. The core of BN involves normalizing the features of each batch by first calculating the mean μ_{batch} and variance $\sigma^2_{\mathrm{batch}}$ for the batch,where m is the number of samples in the batch:

$$\mu_{\mathrm{batch}} = \frac{1}{m} \sum_{i=1}^{m} x_i, \quad \sigma^2_{\mathrm{batch}} = \frac{1}{m} \sum_{i=1}^{m} (x_i - \mu_{\mathrm{batch}})^2 \tag{11}$$

Then, each input x_i is normalized as follows:

$$\hat{x}_i = \frac{x_i - \mu_{\mathrm{batch}}}{\sqrt{\sigma^2_{\mathrm{batch}} + \epsilon}} \tag{12}$$

where ϵ is a small constant to prevent division by zero. Compared to Local Response Normalization (LRN), which only normalizes local regions and may overlook distant feature relationships, BN is more effective in capturing global dependencies. This advantage makes BN particularly beneficial in handling complex and dispersed structural anomalies, as it can better capture features that span across different regions of the image, enhancing model performance.

Another important advantage of BN is that it accelerates model convergence, allowing for higher learning rates while reducing sensitivity to parameter initialization. In complex tasks like structural anomaly detection, BN helps prevent issues such as gradient explosion or vanishing, thus stabilizing the training process. BN further restores the

model's representation capacity through scaling parameter γ and shift parameter β:

$$y_i = \gamma \hat{x}_i + \beta \tag{13}$$

In summary, BN significantly improves the model's generalization ability through global normalization and learnable parameters, making it particularly effective in handling complex and dispersed features.

Results

Data preprocessing and dataset

In this study, nine sequencing datasets were downloaded from the PacBio platform [28], including HG00512, HG00513, HG00514, HG00731, HG00732, HG00733, NA19238, NA19239 and NA19240. These datasets were aligned to the human reference genome GRCh38. Next, we used samtools [29] to sort and index the datasets, generating sorted BAM files. Details of the datasets are listed in Table 1. Then, we used CuteSV (version 2.1.0) to detect structural variants in these BAM files, producing corresponding VCF files. Finally, following the encoding scheme described in the methods section, we converted these data into image form for further analysis. The data distribution for each label (INS, DEL, INV and DUP) is shown in Fig. 3a.

Performance metrics

We used the following evaluation metrics to measure the model's performance:

$$\text{Accuracy} = \frac{TP + TN}{TP + TN + FP + FN} \tag{14}$$

Table 1 Coverage, Average Read Length, and Total Reads for each dataset

Dataset	Coverage[1]	Average Read Length[2]	Total Reads[3]
HG00512	19×	8465.67	8,748,716
HG00513	19×	8886.04	8,055,492
HG00514	40×	8902.38	18,271,333
HG00731	22×	8481.2	11,896,218
HG00732	23×	8156.42	11,863,150
HG00733	44×	8603.6	21,659,549
NA19238	18×	5553.22	12,776,976
NA19239	16×	5395.81	12,384,949
NA19240	37×	5445.1	26,706,454

Coverage: The sequencing depth, indicating how many times each base of the genome is sequenced. Average Read Length: The average length of each sequencing read, measured in base pairs. Total Reads: The total number of individual reads obtained from the sequencing data

$$\text{Precision} = \frac{TP}{TP + FP} \tag{15}$$

$$\text{Recall} = \frac{TP}{TP + FN} \tag{16}$$

$$\text{F1 Score} = 2 \times \frac{\text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}} \tag{17}$$

where TP is True Positives, TN is True Negatives, FP is False Positives, and FN is False Negatives. These metrics provide a comprehensive evaluation of the model's classification ability across different types of structural variations from multiple perspectives.

Optimizing module for enhanced model performance on three datasets

This ablation study aims to evaluate the impact of the multi-head self-attention mechanism (MHSA) and multi-scale convolution (MSC) on model performance. We compared three model configurations: (1) Structural Variations Enhanced AlexNet (SVEA, including MHSA and MSC), (2) The model with MHSA removed but retaining MSC, and (3) The model with MHSA removed and MSC replaced with standard convolutional layers. Experiments were conducted on the HG00514, HG00733, and NA19240 datasets, evaluating each model's performance in terms of accuracy, recall and F1 score (Table 2).

SVEA performed well across all datasets, with accuracy ranging from 97.2% to 97.5% . By combining MHSA and MSC, SVEA demonstrated strong classification performance on different datasets, performing best on the complex HG00733 dataset, where the accuracy reached 97.5%.

When MHSA was removed, the model's performance on HG00514 declined, with accuracy dropping to 96.5%. However, on HG00733, the model's accuracy increased to 97.5% after removing MHSA, indicating that the removal of MHSA had a positive effect on certain datasets. For NA19240, the impact of removing MHSA was minimal, with the model's accuracy remaining nearly unchanged at 97.1% respectively.

When MHSA was further removed and MSC was replaced with standard convolutional layers, the performance improved on the HG00514 and HG00733 datasets, with accuracy reaching 97.1% on HG00514. However, performance on NA19240 significantly declined, with accuracy dropping from 97.1% to 95.8%. These results suggest that although standard convolutional layers may excel in certain scenarios, MSC offers superior feature extraction capabilities for more complex datasets.

Table 2 Performance of SVEA and W/O MHSA, MSC on Different Datasets

Data	Model	DEL			DUP			INS			INV			Overall
		P	R	F1	P	R	F1	P	R	F1	P	R	F1	A(%)
HG00514	SVEA	0.97	0.97	0.97	0.95	0.94	0.95	0.98	0.98	0.98	0.80	0.65	0.72	**97.2**
	w/o SA	0.97	0.97	0.97	0.96	0.92	0.94	0.97	0.98	0.97	0.93	0.62	0.74	96.5
	w/o C	0.96	0.97	0.96	0.95	0.96	0.95	0.98	0.97	0.98	0.00	0.00	0.00	97.1
HG00733	SVEA	0.96	0.98	0.95	0.97	0.95	0.96	0.98	0.96	0.97	0.70	0.72	0.77	**97.5**
	w/o SA	0.96	0.97	0.97	0.97	0.95	0.96	0.98	0.98	0.98	0.91	0.87	0.89	**97.5**
	w/o C	0.95	0.98	0.96	0.93	0.95	0.94	0.98	0.97	0.98	1.00	0.07	0.13	96.8
NA19240	SVEA	0.98	0.96	0.97	0.93	0.97	0.95	0.98	0.99	0.98	0.80	0.67	0.74	**97.2**
	w/o SA	0.96	0.97	0.97	0.96	0.94	0.95	0.98	0.98	0.98	0.92	0.64	0.76	97.1
	w/o C	0.94	0.95	0.95	0.96	0.90	0.93	0.97	0.97	0.97	1.00	0.02	0.04	95.8

P: Precision, *R*: Recall, *F1*: F1 score, *A*: Accuracy. *SVEA*: Structural Variation detection with Enhanced AlexNet architecture. *w/o SA*: Without Multi-Head Self-Attention. *w/o C*: Without Multi-Head Self-Attention and Multi-Scale Convolution.

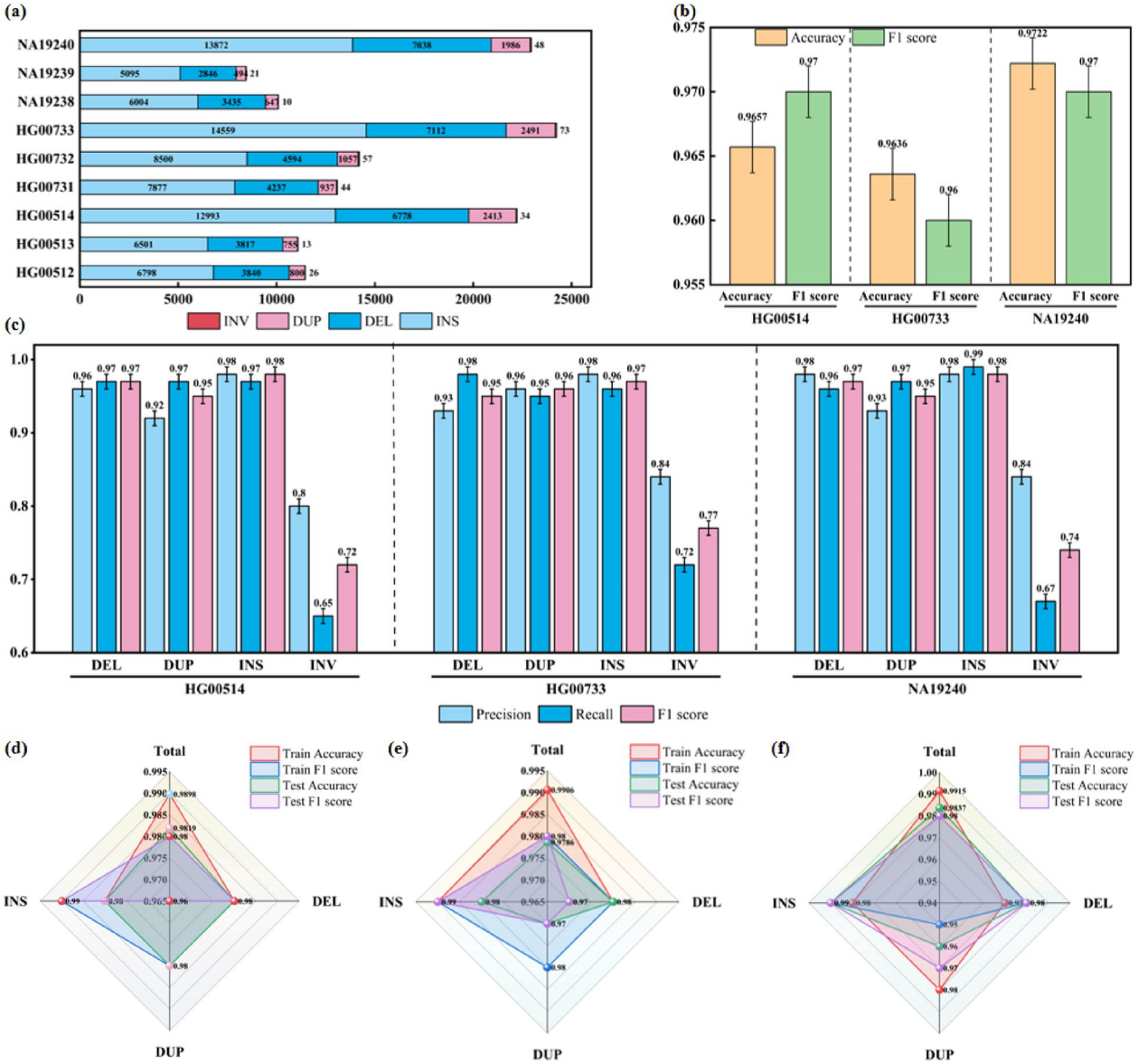

Fig. 3 Performance Evaluation of SVEA Across Various Datasets and Configurations. **a** Detailed composition of each dataset analyzed. **b** Overall accuracy and F1-score results from five-fold cross-validation results across the HG00514, HG00733, and NA19240 datasets. **c** Accuracy, precision, and recall for each label during five-fold cross-validation on the HG00514, HG00733, and NA19240 datasets. **d** Performance results when using HG00514 and HG00733 as the training sets and NA19240 as the validation set. **e** Performance results when using HG00514 and NA19240 as the training sets and HG00733 as the validation set. **f** Performance results when using NA19240 and HG00733 as the training sets and HG00514 as the validation set

As shown in Table 2, For the detection of the DEL type, all model configurations demonstrated consistent performance across the three datasets, with precision ranging from 0.96 to 0.98 and F1 scores from 0.95 to 0.98, indicating consistently high performance. For the DUP type, removing MHSA led to an improvement in precision, particularly in the HG00514 and HG00733 datasets, where precision increased to between 0.96 and 0.97, and F1 scores also improved. However, in the NA19240 dataset, the performance of the model after removing MHSA remained the unchanged or slightly declined.

For the INS type, detection was stable across all configurations, with precision and F1 scores showing minimal variation, consistently ranging between 0.97 and 0.99. The detection of the INV type, however, showed significant fluctuations. The baseline model

performed suboptimally on INV detection, especially in the HG00514 and NA19240 datasets, with F1 scores of 0.72 and 0.74, respectively. After removing MHSA, INV detection performance improved in the HG00733 and NA19240 datasets, with notable increases in precision and F1 scores. However, when MSC was replaced with standard convolutional layers, INV detection performance in HG00514 and NA19240 almost failed, indicating that multi-scale convolution is critical for INV detection feature extraction.

Comparing the ablation study results, SVEA superior detection performance across DEL, DUP, INS, and INV variation types. This was particularly evident in detecting the complex INV type, where MSC played a crucial role. The ablation study reveals that removing MHSA and MSC caused performance declines of varying degrees across multiple variation types. Specifically, the removal of MSC resulted in almost zero precision and F1 scores for the INV type, highlighting the importance of MSC for extracting features in complex variation types. Similarly, the inclusion of MHSA improved the detection accuracy of DEL and DUP variations and ensured high recall and F1 scores. Therefore, the combination of MHSA and MSC significantly enhanced the model's performance in handling diverse structural variations, providing strong robustness and accuracy across all datasets.

In the early stages of model optimization, we conducted some preliminary ablation experiments, including replacing the global average pooling layer with a fully connected layer or omitting the batch normalization layer. The experimental results indicated that, under the same hyperparameters, these modifications led to a significant decrease in detection performance, with almost no effective detections, and only a small portion of insertions were identified. Furthermore, when we adjusted the hyperparameters to bring the model's accuracy closer to that of the SVEA model, the model's convergence speed was slower, with convergence time being 1x to several times longer than that of the SVEA model. These results suggest that both the global average pooling layer and batch normalization play a crucial role in accelerating convergence and improving accuracy within the model.

Performance analysis of SVEA via five-fold cross-validation on three datasets

In this study, to comprehensively evaluate the model's generalization capability across different types of structural variations, we performed five-fold cross-validation on three datasets (HG00514, HG00733, NA19240). The five-fold cross-validation process involved splitting each dataset into five subsets, with one subset used as the validation set and the remaining four subsets used for training. This process was repeated five times, and the average result across these experiments represented the model's overall performance on the dataset. This approach allowed us to thoroughly assess the model's classification performance across different structural variation types (INS, DEL, DUP, INV).

Given that the number of INV samples in each dataset was significantly lower than other types of structural variations (such as INS and DEL), we combined INV samples from nine datasets for five-fold cross-validation. This approach aimed to address the issue of insufficient INV samples, ensuring that the model encountered a sufficient number of INV samples during training and validation. This strategy enhanced the model's ability to learn and detect this type of variation, improving its generalization and detection performance.

As shown in Fig. 3b, the model's overall performance across the three datasets HG00514, HG00733, and NA19240 was consistent. The average accuracy for HG00514 was 96.57% with an F1-score of 0.97. Similarly, HG00733 had an average accuracy of 96.36% with an F1-score of 0.96, while NA19240 showed an accuracy of 97.22% and an F1-score of 0.97. Overall, the model demonstrated high accuracy and F1-scores across these three datasets, indicating strong generalization capability when handling different types of structural variations.

As illustrated in Fig. 3c, the model performed exceptionally well on INS and DEL types, with F1-scores close to 0.98, showing that the model could effectively identify and detect these types while maintaining a good balance between precision and recall. Although the F1-scores for the DUP type were generally high (ranging from 0.93 to 0.96), some fluctuations were observed. In contrast, the model's performance on the INV type was noticeably lower than for the other types, with F1-scores of 0.72 (HG00514), 0.77 (HG00733), and 0.74 (NA19240). Particularly in some folds, the recall for the INV type was zero, resulting in a significant drop in the F1-score, likely due to the limited number of INV samples affecting the model's generalization capability.

In summary, our approach exhibited strong performance across multiple types of structural variations, especially for INS and DEL, consistently achieving high F1 scores and stable results across datasets. By employing five-fold cross-validation and combining datasets to mitigate the scarcity of INV samples, we significantly enhanced the model's generalization capability in detecting complex variations.

Performance analysis of SVEA via independent validation on nine datasets

Cross-dataset cross-validation is designed to comprehensively evaluate the generalization capability of the model, particularly in terms of its adaptability across different

datasets and structural variation types. This validation approach allows for an in-depth assessment of the model's stability and performance variations when applied to data from diverse sources, providing a more accurate reflection of its practical applicability and broad utility. In this part of the experiments, we divided the cross-dataset cross-validation into two stages. Subsequently, we will present and discuss the specific results of these validations and conduct a detailed analysis of the model's generalization performance.

Cross-dataset performance analysis of SVEA

In the first stage, we used two out of the three datasets (HG00514, HG00733, and NA19240) for model training, while the remaining dataset was used for validation. This cross-validation approach aimed to evaluate the model's generalization capability on unseen datasets. By alternately using two of the three datasets for training and one for validation, we were able to thoroughly test the model's adaptability and stability in a cross-dataset setting.

The results from the first stage, shown in Fig. 3d–f, indicate that the model's performance is consistent across different datasets, especially for the DEL, DUP, and INS structural variation types, where the training and testing F1-scores are consistently close to or above 0.98, demonstrating exceptional stability. The INS type, in particular, frequently achieved an F1-score of 0.99. This suggests that the model can effectively learn the features of these variation types and generalize well to unseen datasets.

Specifically, when HG00514 and HG00733 were used as the training set and NA19240 as the testing set, the model achieved an accuracy of 98.19% and an F1-score of 0.98 on the testing set. When HG00514 and NA19240 were used for training and HG00733 for testing, the testing set accuracy was 97.86% with an F1-score of 0.98. In the experiment where HG00733 and NA19240 were used for training and HG00514 for testing, the testing set accuracy reached 98.37%, and the F1-score was also 0.98. Overall, the model's performance across different testing sets was relatively balanced, demonstrating strong generalization ability.

However, due to the limited number of INV samples, the model's performance on this variation type showed significant fluctuations, making the results less reliable. Therefore, we do not discuss the INV results in detail. In the future, increasing the number of INV samples, applying data augmentation techniques, or using weighted loss functions could improve the model's performance on INV types.

Performance analysis of SVEA on six independent datasets

In the second stage, we combined HG00514, HG00733, and NA19240 as the training set, using six other datasets for validation. This setup allowed us to test the model's performance on a more diverse range of data and assess its generalization ability when trained on multiple datasets. Since the sample distribution and characteristics of these six validation sets may significantly differ from the training set, cross-dataset validation reveals whether the model remains stable when faced with complex and non-uniformly distributed data.

For validation, six datasets were used: HG00512, HG00513, HG00731, HG00732, NA19238, and NA19239. The experiment primarily evaluated overall accuracy, F1 score, recall, as well as the precision, F1 score, and recall for the four main types of structural variations (DEL, DUP, INS and INV). In Fig. 4a, the results showed that the overall accuracy across all datasets ranged between 98.87% and 99.06%, with overall F1 scores and recall values both at 0.99, indicating that the model has good generalization capability across different datasets.

For the four types of structural variations, as shown in Fig. 4b–d, the DEL type performed the best, with all datasets achieving a precision, F1 score, and recall of 0.99, demonstrating the model's excellent performance in detecting deletions. The detection of DUP types showed slight variation across different datasets, with precision ranging between 0.96 and 0.98, and F1 scores and recall values mostly around 0.98, indicating that the model maintained stability in identifying duplications, although performance slightly declined in the HG00731 and NA19238 datasets. The performance for INS types was also quite stable, with precision, F1 scores, and recall values close to 0.99 across all datasets, reflecting the model's high accuracy in insertion detection.

In contrast, the performance for INV types was weaker and more variable. Particularly in the HG00512 and HG00513 datasets, precision and F1 scores were lower, at 0.83 and 0.78, and 0.89 and 0.73, respectively, indicating the model's limitations in detecting inversions. In the NA19238 and NA19239 datasets, the F1 score and recall dropped further to 0.36 and 0.72, showing that the model struggled with inversion detection, especially in datasets with fewer samples. This could be due to the imbalance or complexity of inversion features.

In summary, the model performed excellently in detecting the common structural variation types (DEL, DUP, INS), demonstrating strong stability and generalization ability. However, detecting inversion types (INV) remains a significant challenge. Future improvements should focus on data augmentation and simulated data to further enhance the model's overall performance across different datasets.

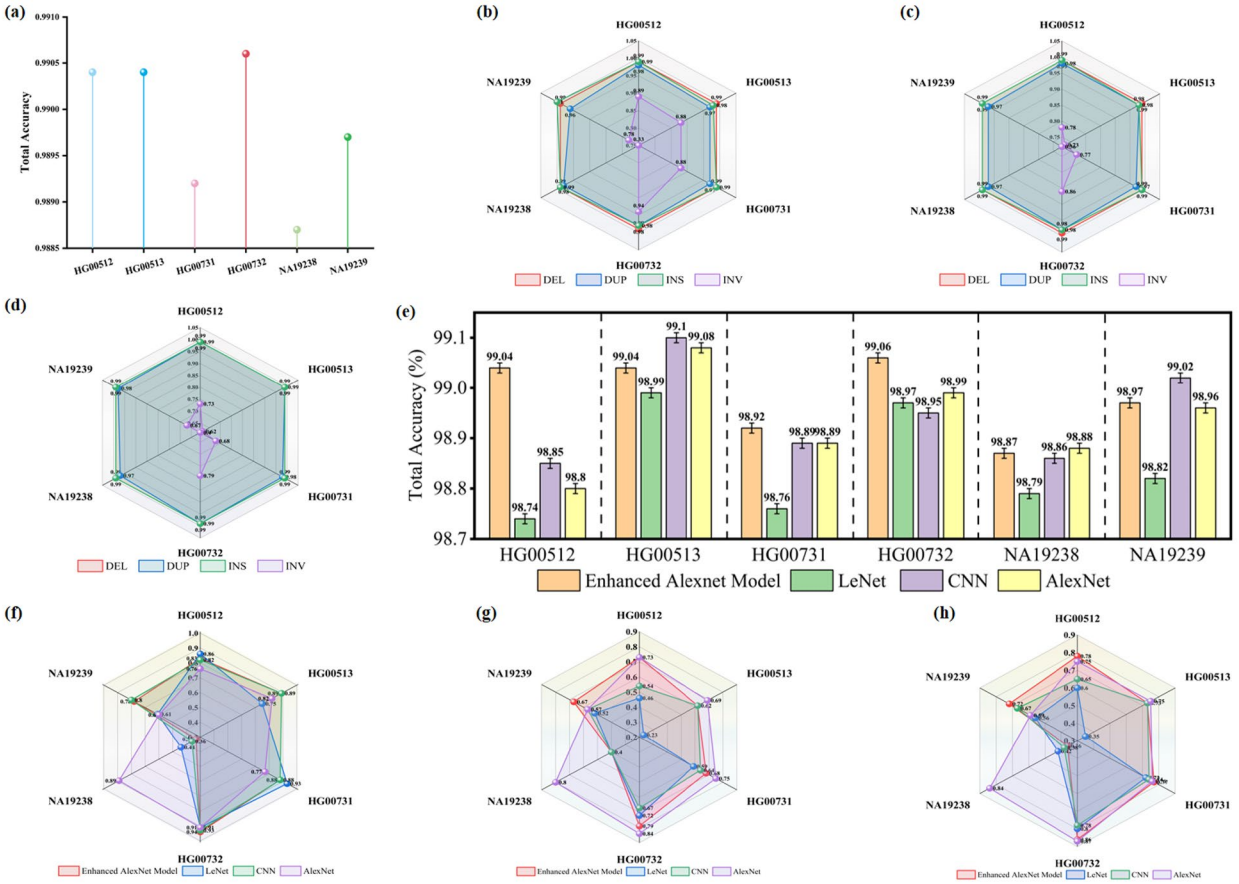

Fig. 4 Performance Analysis of SVEA on Six Independent Datasets (**a**) Accuracy across the six datasets. **b** Precision for each label on the six datasets. **c** F1 score for each label on the six datasets. **d** Recall for each label on the six datasets. **e** Accuracy comparison of multiple models across different datasets. **f** Precision of multiple models for the INV label. **g** Recall of multiple models for the INV label. **h** F1 score of multiple models for the INV label

Comparison of SVEA with other models

To further demonstrate the performance of SVEA, we try to compare it with other deep learning-based structural variation (SV) detection methods, including SVision [17], SVcnn [30], and cnnLSV [31]. However, since the complete code for these models is unavailable, we were unable to run them directly on our dataset. Instead, we constructed the models based on descriptions in the papers and implemented them on our dataset. Despite this, these models represent different architectural approaches and design philosophies, such as SVision using AlexNet, SVcnn utilizing LeNet, and cnnLSV implementing a convolutional neural network. As a result, we compare our method with these models based on their respective deep learning architectures and evaluate their performance accordingly.

The dataset was divided into a training set and a validation set: the training set comprised HG00514, HG00733, and NA19240 for model training and parameter optimization, while the validation set consisted of HG00512, HG00513, HG00731, HG00732, NA19238, and NA19239 to evaluate the models' generalization capability on unseen data. By comparing the performance of these models on the validation set, we aimed to reveal their differences in accuracy and other key metrics, thereby assessing the potential advantages of our own model in this task and providing clear guidance for further optimization.

As shown in Fig. 4e, the Enhanced AlexNet model achieved consistently high accuracy across all six datasets, with the best performance on the HG00512, HG00731, and HG00732 datasets. In terms of F1 score, all models performed similarly across different datasets, achieving 0.99, indicating a uniform classification balance.

For each label, all models achieved precision, F1 score, and recall of 0.99 on the INS and DEL labels, further demonstrating stable performance on these labels. For the DUP label, the precision, F1 score, and recall were relatively close across models, with the Enhanced

AlexNet model performing best, achieving between 0.98 and 0.99 in all three metrics. The other models (LeNet, CNN, and AlexNet), though slightly behind, maintained stable performance on the DUP label, with precision and F1 scores mostly above 0.97, indicating consistent classification performance across models on the DUP label.

In the subsequent analysis, we focused primarily on the transitional segment of the INV label. This is due to the considerable variability in classification performance on the INV label, with significant differences observed across models. Further analysis of the INV label helps us better understand model stability across different datasets and sample sizes, particularly when dealing with limited samples and complex structural variations. As shown in Fig. 4f–h, the Enhanced AlexNet model stood out on certain datasets, achieving a maximum precision of 0.94, along with high F1 score and recall. The performance of other models fluctuated significantly on the INV label, with LeNet and AlexNet showing the lowest precision, dropping to 0.33 and 0.36, and F1 scores and recall also decreasing to 0.35 and 0.4, respectively, indicating weaker stability when handling the INV label. Comparatively, the Enhanced AlexNet showed the most stable performance on the INV label, followed by AlexNet and CNN, while LeNet exhibited the lowest classification accuracy and consistency. On the NA19238 dataset, the four models (the Enhanced AlexNet, LeNet, CNN, AlexNet) show significant differences in classification performance on the INV label, especially given the small sample size of only 10. Although the Enhanced AlexNet has lower precision (0.33), recall (0.4), and F1 score (0.36) compared to AlexNet on this dataset, this can be attributed to the extremely limited number of INV samples, which leads to greater variability in model training and evaluation outcomes.

Overall, the Enhanced AlexNet model demonstrated higher accuracy and consistency in classifying the DUP and INV labels, especially showing significant advantages on the more challenging INV label. This result suggests that Enhanced AlexNet possesses superior adaptability and robustness in multi-label classification tasks.

Conclusion

This study presents and validates SVEA, a deep learning model designed for structural variation prediction. SVEA integrates a multi-head self-attention mechanism and a multi-scale convolutional module within the traditional AlexNet architecture, significantly enhancing the model's ability to capture global context and multi-scale features. By employing an innovative multi-channel image encoding method based on CIGAR strings, SVEA effectively leverages alignment information, improving detection accuracy and generalization across different genomic regions. This approach not only extracts valuable features from alignment data but also processes the data in an image-based manner, further enhancing the model's understanding and handling of complex variations.

Experimental results show that SVEA outperforms existing SV prediction models in terms of accuracy and F1 score, highlighting the unique advantages of combining advanced deep learning architectures with innovative image encoding methods. This combination enables the model to achieve higher stability and precision in detecting a wide range of structural variations, particularly in detecting complex genomic alterations.

Despite these promising results, challenges remain, particularly in predicting rare variations and complex genomic regions. Future research will focus on further optimizing SVEA's architecture to enhance its ability to predict rare variations and improve the encoding method to better differentiate between duplications and insertions, reducing reliance on predefined variant types. Additionally, future work will focus on improving the model's accuracy and robustness when handling more complex and fine-grained structural variations, aiming to achieve broader applicability.

Acknowledgments
Not applicable.

Author contributions
TQ and JL conceived and designed the experiments; TQ performed the experiments and analyzed the data; TQ, YG and LJ wrote the paper. LJ and JT supervised the experiments and reviewed the manuscript. All authors read and approved the final manuscript.

Funding
This work is supported by the Shenzhen KQTD Project under grant no. KQTD20200820113106007; This work is also supported by a grant from the National Key R&D Program of China (2020YFA0908400), the National Natural Science Foundation of China (NSFC 62172296).

Availability of data and materials
All codes associated with this study are available at https://github.com/Qiutaxx/SVEA. All datasets were downloaded from https://ftp.1000genomes.ebi.ac.uk/vol1/ftp/data_collections/hgsv_sv_discovery/working/20160905_smithm_pacbio_aligns/.

Declarations

Ethics approval and consent to participate
Not applicable.

Consent for publication
Not applicable.

Competing interests
No competing interest is declared.

Received: 25 November 2024 Accepted: 6 February 2025
Published online: 22 February 2025

References

1. Feuk L, Carson AR, Scherer SW. Structural variation in the human genome. Nat Rev Genetics. 2006;7(2):85–97.
2. Garcia-Prieto CA, Martínez-Jiménez F, Valencia A, Porta-Pardo E. Detection of oncogenic and clinically actionable mutations in cancer genomes critically depends on variant calling tools. Bioinformatics. 2022;38(12):3181–91.
3. Singleton A, Farrer M, Johnson J, Singleton A, Hague S, Kachergus J, Hulihan M, Peuralinna T, Dutra A, Nussbaum R, et al. α-synuclein locus triplication causes parkinson's disease. Science. 2003;302(5646):841–841.
4. Rovelet-Lecrux A, Hannequin D, Raux G, Meur NL, Laquerrière A, Vital A, Dumanchin C, Feuillette S, Brice A, Vercelletto M, et al. App locus duplication causes autosomal dominant early-onset alzheimer disease with cerebral amyloid angiopathy. Nat Genetics. 2006;38(1):24–6.
5. Fiksinski AM, Hoftman GD, Vorstman JA, Bearden CE. A genetics-first approach to understanding autism and schizophrenia spectrum disorders: the 22q11. 2 deletion syndrome. Mol Psychiatry. 2023;28(1):341–53.
6. Cardoso AR, Lopes-Marques M, Silva RM, Serrano C, Amorim A, Prata MJ, Azevedo L. Essential genetic findings in neurodevelopmental disorders. Human Genomics. 2019;13:1–7.
7. Feuk L. Inversion variants in the human genome: role in disease and genome architecture. Genome Med. 2010;2:1–8.
8. Jiang T, Liu Y, Jiang Y, Li J, Gao Y, Cui Z, Liu Y, Liu B, Wang Y. Long-read-based human genomic structural variation detection with cutesv. Genome Biol. 2020;21:1–24.
9. Sedlazeck FJ, Rescheneder P, Smolka M, Fang H, Nattestad M, Von Haeseler A, Schatz MC. Accurate detection of complex structural variations using single-molecule sequencing. Nat Methods. 2018;15(6):461–8.
10. Heller D, Vingron M. Svim: structural variant identification using mapped long reads. Bioinformatics. 2019;35(17):2907–15.
11. Chen Y, Wang AY, Barkley CA, Zhang Y, Zhao X, Gao M, Edmonds MD, Chong Z. Deciphering the exact breakpoints of structural variations using long sequencing reads with debreak. Nat Commun. 2023;14(1):283.
12. Li H. Minimap2: pairwise alignment for nucleotide sequences. Bioinformatics. 2018;34(18):3094–100.
13. Li R, Zhu H, Ruan J, Qian W, Fang X, Shi Z, Li Y, Li S, Shan G, Kristiansen K, et al. De novo assembly of human genomes with massively parallel short read sequencing. Genome Res. 2010;20(2):265–72.
14. Ahsan MU, Liu Q, Perdomo JE, Fang L, Wang K. A survey of algorithms for the detection of genomic structural variants from long-read sequencing data. Nat Methods. 2023;20(8):1143–58.
15. Luo J, Ding H, Shen J, Zhai H, Wu Z, Yan C, Luo H. Breaknet: detecting deletions using long reads and a deep learning approach. BMC Bioinform. 2021;22:1–13.
16. Ding H, Luo J. Mamnet: detecting and genotyping deletions and insertions based on long reads and a deep learning approach. Brief Bioinform. 2022;23(5):195.
17. Lin J, Wang S, Audano PA, Meng D, Flores JI, Kosters W, Yang X, Jia P, Marschall T, Beck CR, et al. Svision: a deep learning approach to resolve complex structural variants. Nat Methods. 2022;19(10):1230–3.
18. Popic V, Rohlicek C, Cunial F, Hajirasouliha I, Meleshko D, Garimella K, Maheshwari A. Cue: a deep-learning framework for structural variant discovery and genotyping. Nat Methods. 2023;20(4):559–68.
19. Poplin R, Chang PC, Alexander D, et al. A universal snp and small-indel variant caller using deep neural networks. Nat Biotechnol. 2018;36(10):983–7.
20. Bhattacharya, S., Shaw, V., Singh, P.K., et al.: Sv-net: a deep learning approach to video based human activity recognition. In: Proceedings of the 11th International Conference on Soft Computing and Pattern Recognition (SoCPaR 2019), pp. 10–20. Springer (2021)
21. Wu Y, Tian L, Pirastu M, et al. MATCHCLIP: locate precise breakpoints for copy number variation using CIGAR string by matching soft clipped reads. Front Genet. 2013;4:157.
22. Ashish V. Attention is all you need. Adv Neural Inform Proc Syst. 2017;30:I.
23. Szegedy, C., Liu, W., Jia, Y., Sermanet, P., Reed, S., Anguelov, D., Erhan, D., Vanhoucke, V., Rabinovich, A.: Going deeper with convolutions. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 1–9 (2015)
24. Wang, H., Tu, M.: Enhancing attention models via multi-head collaboration. In: 2020 International Conference on Asian Language Processing (IALP), pp. 19–23 (2020). IEEE
25. Szegedy, C., Liu, W., Jia, Y., Sermanet, P., Reed, S., Anguelov, D., Erhan, D., Vanhoucke, V., Rabinovich, A.: Going deeper with convolutions. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 1–9 (2015)
26. Ioffe, S.: Batch normalization: Accelerating deep network training by reducing internal covariate shift. arXiv preprint arXiv:1502.03167 (2015)
27. Lin, M.: Network in network. arXiv preprint arXiv:1312.4400 (2013)
28. Wenger AM, Peluso P, Rowell WJ, Chang P-C, Hall RJ, Concepcion GT, Ebler J, Fungtammasan A, Kolesnikov A, Olson ND, et al. Accurate circular consensus long-read sequencing improves variant detection and assembly of a human genome. Nat Biotechnol. 2019;37(10):1155–62.
29. Li H, Handsaker B, Wysoker A, Fennell T, Ruan J, Homer N, Marth G, Abecasis G, Durbin R, Subgroup GPDP. The sequence alignment/map format and samtools. Bioinformatics. 2009;25(16):2078–9.
30. Zheng Y, Shang X. Svcnn: an accurate deep learning-based method for detecting structural variation based on long-read data. BMC Bioinform. 2023;24(1):213.
31. Ma H, Zhong C, Chen D, He H, Yang F. cnnlsv: detecting structural variants by encoding long-read alignment information and convolutional neural network. BMC Bioinform. 2023;24(1):119.

Publisher's Note