Computational and Structural Biotechnology Journal 23 (2024) 3430–3444

Contents lists available at ScienceDirect

Computational and Structural Biotechnology Journal

journal homepage: www.elsevier.com/locate/csbj

Research article

# BaseNet: A transformer-based toolkit for nanopore sequencing signal decoding

Qingwen Li [a,b], Chen Sun [c], Daqian Wang [c,*], Jizhong Lou [a,b,c,**]

[a] Key Laboratory of Epigenetic Regulation and Intervention, Center for Excellence in Biomacromolecules, Institute of Biophysics, Chinese Academy of Sciences, Beijing 100101, China
[b] University of Chinese Academy of Sciences, Beijing 100049, China
[c] Beijing Polyseq Biotech Co. Ltd., Beijing 100089, China

ARTICLE INFO

*Keywords:*
Nanopore sequencing
Basecall
Transformer
Machine learning algorithm

ABSTRACT

Nanopore sequencing provides a rapid, convenient and high-throughput solution for nucleic acid sequencing. Accurate basecalling in nanopore sequencing is crucial for downstream analysis. Traditional approaches such as Hidden Markov Models (HMM), Recurrent Neural Networks (RNN), and Convolutional Neural Networks (CNN) have improved basecalling accuracy but there is a continuous need for higher accuracy and reliability. In this study, we introduce BaseNet (https://github.com/liqingwen98/BaseNet), an open-source toolkit that utilizes transformer models for advanced signal decoding in nanopore sequencing. BaseNet incorporates both autoregressive and non-autoregressive transformer-based decoding mechanisms, offering state-of-the-art algorithms freely accessible for future improvement. Our research indicates that cross-attention weights effectively map the relationship between current signals and base sequences, joint loss training through adding a pair of forward and reverse decoder facilitate model converge, and large-scale pre-trained models achieve superior decoding accuracy. This study helps to advance the field of nanopore sequencing signal decoding, contributes to technological advancements, and provides novel concepts and tools for researchers and practitioners.

## 1. Introduction

Nanopore sequencing has emerged as a pivotal technology in modern genomics, offering fast, convenient and high-throughput sequencing for both DNA and RNA. This technique involves passing a DNA or RNA strand through a nanopore and measuring changes in ionic current to determine the sequence of nucleotides. Due to its speed, cost-effectiveness, and portability, nanopore sequencing has been widely applied in genomics research, medical diagnostics, personalized medicine, and other related fields. In 2018, Jain et al. [1] developed an ultra-long read length method using ONT's CsgG mutant R.9.4.1, successfully de novo sequencing the reference human genome from the GM12878 cell line. This method achieved 30-fold genome coverage and an impressive 99.88 % sequence accuracy, comparable to other short- and long-read platforms. Additionally, Davenport et al. [2] employed nanopore sequencing to map 5-methylcytosine signals across the genome, integrating these findings with gene transcription profiles of regenerative liver and primary hepatocellular carcinoma. Their work highlighted glucokinase as a tumor suppressor gene, which inhibits liver cancer cell proliferation by inducing lactate accumulation. The importance of rapid pathogen detection in clinical settings is underscored by the use of ONT MinION sequencers in tracking multiple outbreaks, including Ebola [3] and influenza [4] in Guinea, Zika in Brazil and the United States [5], and SARS-CoV-2 Alpha and Delta variants in Ukraine [6]. Furthermore, Boykin et al. [7] utilized MinION for metagenomic sequencing to detect cassava mosaic virus infection in cassava plants, completing the entire process—from sample collection to sequencing analysis—within 3 h at a cassava farm in the southern Sahara Desert.

Accurate decoding of nucleic acid sequences from the current signal change is crucial for effective downstream analysis in nanopore sequencing. These signal changes are complex, necessitating the use of artificial intelligence-based algorithms to decode the base sequence accurately [8]. However, raw electrical signals are often influenced by various factors including system noise, base drift, and the heterogeneity

* Corresponding author.
** Corresponding author at: Key Laboratory of Epigenetic Regulation and Intervention, Center for Excellence in Biomacromolecules, Institute of Biophysics, Chinese Academy of Sciences, Beijing 100101, China.
*E-mail addresses:* daqian.wang@polyseq.com (D. Wang), jlou@ibp.ac.cn (J. Lou).

https://doi.org/10.1016/j.csbj.2024.09.016
Received 18 June 2024; Received in revised form 18 September 2024; Accepted 24 September 2024
Available online 25 September 2024

of through-pore velocities etc. Consequently, achieving high-precision signal decoding in nanopore sequencing is a challenging yet essential for its widespread application and reliability [9].

Nanopore sequencing and speech recognition technology share several fundamental characteristics, as both require the effective modeling of temporal dependencies within noisy, variable-length signals and the transformation of continuous, complex data into meaningful sequences—whether those sequences are nucleotide bases or spoken words. While the development trajectory of nanopore basecallers follows a similar path to that of speech recognition models, it is important to note that the progress in nanopore basecaller development has significantly lagged behind advancements in speech recognition [10, 11]. Previous studies have predominantly employed basecallers constructed using methods such as Hidden Markov Models (HMM) [12], Recurrent Neural Networks (RNN), and Convolutional Neural Networks (CNN) [13-15] to decode the raw electrical signals in nanopore sequencing. While these methods have achieved significant success, they also have limitations. HMMs require manual feature extraction from signals and have limited capabilities for modeling long-range dependencies [16]. RNNs can model long-range dependencies but suffer from issues such as vanishing and exploding gradients. CNNs, on the other hand, struggle to capture temporal information in sequences. These limitations restrict the decoding accuracy and reliability of some existing basecallers.

To address these challenges, the transformer model was originally proposed for machine translation tasks [17], leveraging self-attention and multi-head attention mechanisms to capture long-range dependencies in sequences. It has since achieved remarkable milestones in natural language processing, image recognition [18-20], and automatic speech recognition [21,22], demonstrating powerful capabilities in biology as well, including protein generation, structure prediction, drug design, and biological sequence recognition [23-25]. Transformer's ability to effectively model long-range dependencies makes it a promising tool for many applications. However, in the field of nanopore sequencing, the application and development of transformer remains limited. Recent efforts have only utilized the transformer encoder and connectionist temporal classification (CTC) decoder for basecalling [26], leaving much potential for further exploration and advancement.

Here, we introduce BaseNet, an open-source toolkit that provides a transformer-based advanced algorithm platform for signal decoding in nanopore sequencing. BaseNet features several key components: 1). Autoregressive decoding: a transformer model using beam search for enhanced accuracy; 2). Non-autoregressive decoding: a transformer with a rescore decoding mechanism, trained using a combination of CTC and attention-based encoder-decoder (AED) [27]; 3). Paraformer: a non-autoregressive decoder employing a Continuous Integrate-and-Fire (CIF) based predictor and a glancing language model (GLM) based generator [28]; 4). Large-scale pre-trained model: a model fine-tuned using contrastive learning and diversity learning for improved performance on nanopore sequencing data [29]; 5). Conditional random field (CRF) model: refined by a linear complexity attention mechanism to enhance decoding efficiency [30].

To our knowledge, BaseNet is the first to introduce transformer-based autoregressive and non-autoregressive decoding mechanisms in nanopore sequencing. Importantly, BaseNet provides several state-of-the-art transformer algorithms publicly available for for free access and further improvement. Below we outlines the main algorithm principles and model structures within BaseNet, as well as experimental results on key benchmark datasets of nanopore sequencing. Our results demonstrates that BaseNet achieves competitive performance, comparable to recent basecallers from Oxford Nanopore Technologies (ONT). The cross-attention weights in the transformer model effectively map the relationship between current signals and base sequences, joint loss training by adding forward and reverse decoders on the top of model improves convergence, and large-scale pre-trained model achieves higher decoding accuracy. Our findings also reveal the existence of common 'generic features' between speech waveforms and sequencing signals in model representation, highlighting the versatility and robustness of BaseNet in handling diverse signal types.

Thus, our studies advance the development of nanopore sequencing signal decoding, contribute to technological improvement, and provide novel ideas and valuable references for the field.

## 2. Materials and methods

The raw current signals generated by nanopore sequencing are fed into BaseNet, which decodes the base sequences using its advanced decoding methods.

### 2.1. Benchmark dataset

The benchmark dataset used in this study was proposed by Wick et al. and has been widely adopted for training and testing multiple basecallers [31]. It consists of both a training set and a test set. The training set includes genomes from fifty species, comprising 30 Klebsiella pneumoniae genomes, 10 Enterobacteriaceae genomes, and 10 Proteobacteria genomes. The test set contains genomes from 10 species, including 4 Klebsiella pneumoniae genomes, as well as genomes from Acinetobacter pittii, Haemophilus haemolyticus, Serratia marcescens, Shigella sonnei, Staphylococcus aureus, and Stenotrophomonas maltophilia. Each dataset includes raw current signals along with their corresponding contig sequences. To ensure the accuracy of the label sequences, Wick et al. performed second-generation sequencing and assembly to obtain the contig sequences. The proportion of various bases in the dataset and the distribution statistics of data size and read length of each species are shown in the Fig. 1.

Before model training, the original signal underwent preprocessing, which included normalization and segmentation.

Initially, the signals were evaluated to determine a noise threshold, computed as follows:

$$threshold = \frac{\sigma_{signal}}{threshold\_factor}$$

Where the threshold factor was set to 6.0 in this study.

This threshold served as a baseline for identifying sections of the signal with higher-than-average noise levels. The signals were then divided into non-overlapping segments of a specified length, set to 100 in this study. The standard deviation of each segment was compared against the threshold, and segments with a standard deviation exceeding this threshold were flagged as noisy.

To identify continuous noisy regions, peak detection was performed on the noise array. A peak-finding algorithm was employed to locate the widest peak, representing the longest continuous noisy section. If such a section was identified, it was used to compute the median and median absolute deviation (MAD); otherwise, the entire signal was utilized for median and MAD calculation. The MAD was calculated as follows:

$$MAD = median(|x - median(x)|) \times factor$$

Where factor was set to 1.4826 in this study.

Finally, the normalized signals were obtained using the following formula:

$$normalized\_signal = \frac{signal - median}{MAD}$$

Due to the long read lengths in nanopore sequencing, each current signal is exceptionally lengthy. For the model's training and validation, we divided the original current signals into matrices with a maximum length of 10,000 sampling points to enhance training efficiency. To ensure accurate base sequences for each fragment and effective training, we first performed basecalling on the electrical signals using Guppy [32]. The resulting sequences were then aligned to the reference

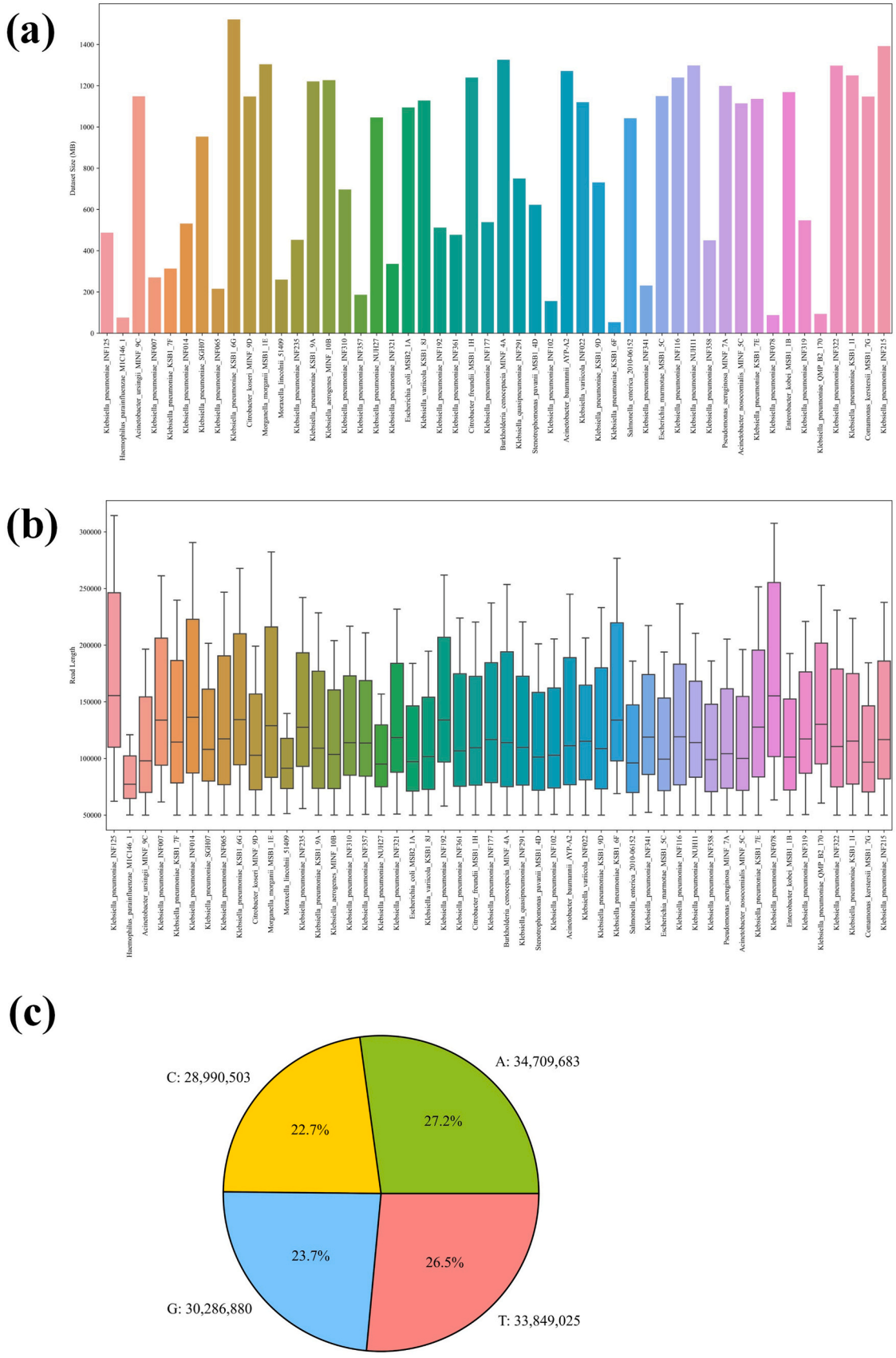


**Fig. 1.** Summary of benchmark dataset characteristics. (a) Data size statistics of each species in the dataset. (b) Read length distribution statistics of each species in the dataset. (c) Statistics of different bases proportion in the dataset.

genome, and only when mapping the alignment accuracy exceeded 60 %, the signal and its corresponding reference genome segment were retained for the training and validation sets. The 60 % threshold was selected based on preliminary experiments conducted using a 5-layer LSTM model with a CTC decoder. These experiments demonstrated that deviations from this threshold—whether increasing or decreasing it—result in a decrease in model accuracy. A higher threshold reduces the number of sequences, thereby limiting the diversity of the training data, while a lower threshold includes sequences with major alignment errors, introducing noise into the model training.

### 2.2. Self-attentionand fast-attention mechanism

The self-attention mechanism enables the model to simultaneously attend to all positions in the sequence, capturing global dependencies. This mechanism also automatically calculates importance weights for each position based on the input sequence, allowing the model to focus on meaningful positions relevant to the current task. Unlike traditional fixed-weight mechanisms, self-attention provides greater flexibility to adapt to various tasks and inputs. The computation is as follows:

$$(Q, K, V) = X(W^Q, W^K, W^V)$$

$$Attention(Q, K, V) = softmax(\frac{QK^T}{\sqrt{d_k}})V$$

Here, $X$ is the input matrix, ($W^Q$, $W^K$, $W^V$) are three trainable parameter matrices. $Q$, $K$, and $V$ are the query, key, and value matrices, respectively, obtained by linear transformations of the input matrix. $d_k$ denotes the dimension of the key.

The multi-head self-attention mechanism allows the model to learn multiple different attention representations simultaneously. Each attention head can focus on different parts of the sequence, providing independent expressive capabilities. This approach captures semantic information at various levels, enhancing the model's ability to comprehend the input sequence and improve its representation and generalization capabilities. And the computation follows:

$$\mathrm{MultiHead}(\mathrm{Q}, \mathrm{K}, \mathrm{V}) = \mathrm{Concat}(\mathrm{head}_1, \ldots, \mathrm{head}_h)W^O$$

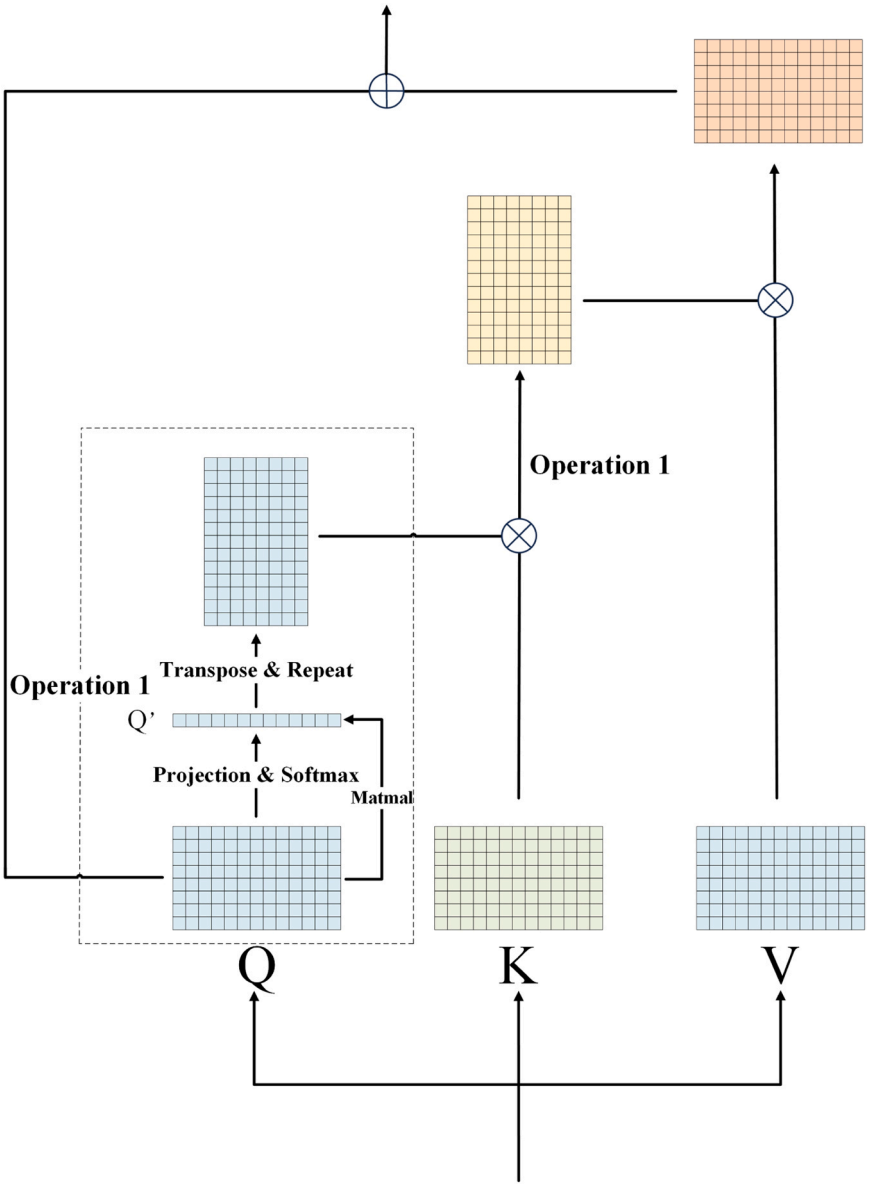


**Fig. 2.** Fast attention mechanism in BaseNet. This schematic illustrates the fast attention mechanism used in BaseNet. The method involves training separate matrices, $\alpha$ and $\beta$, each with dimensions [$N$, 1], for $Q$ and $K$, respectively. These matrices are used to perform a weighted summations on $Q$ and $K$, resulting in transformed matrices $Q$' and $K$'. The transformed matrices are then multiplied together, reducing the computational complexity to $O(N{\cdot}d)$.

$$\text{head}_i = Attention(QW_i^Q, KW_i^K, VW_i^V)$$

Where $W^{\circ}$ is a trainable parameter matrix.

The algorithmic complexity of the attention mechanism described above is $O(N^2 \cdot d)$, where $N$ represents the sequence length and $d$ denotes the dimensionality of the matrices, resulting in a quadratic increase in computational efficiency relative to the sequence length. To alleviate computational costs, Wu et al. proposed a fast attention mechanism with linear complexity[30]. This method involves training separate matrices $\alpha$ and $\beta$, both with dimensions [$N$, 1], for $Q$ and $K$, respectively. These matrices are used to perform a weighted summation on $Q$ and $K$, resulting in transformed matrices $Q$' and $K$'. Subsequently, the transformed matrices are multiplied together, reducing the complexity to $O(N \cdot d)$ [30]. The fast attention mechanism we have developed is illustrated in Fig. 2.

### 2.3. Autoregressive transformer and beam search

The model in this study draws inspiration from the transformer model initially proposed by the Google Machine Translation team[17], with specific adaptations tailored for nanopore sequencing data as shown in Fig. 3. It primarily comprises convolutional modules for feature extraction and down-sampling, an encoder for context modeling to generate hidden vectors, and a decoder for predicting the next time step output based on hidden vectors and generated sequences. Both the encoder and decoder modules consist of 8 layers. The model employs the self-attention mechanism in both the encoder and decoder stages.

In the inference stage, to enhance prediction accuracy, we combine autoregressive decoding with a beam search strategy. A beam size of 4 is selected, meaning that at each time step $t$ during inference, the model retains the top 4 sequences with the highest scores. These sequences are then used as input for predicting the sequences at time step $t+1$ (i.e., retains top sequences with the highest scores among the 16 candidate sequences). The process is repeated iteratively until the decoding is complete. Finally, the sequence with the highest cumulative score across all time steps is selected as the decoding result.

### 2.4. Pre-training and fine-tuning based on large-scale model

Baevski et al. previously introduced wav2vec2.0 [29], a self-supervised pre-training method that has shown significant success in speech recognition. Inspired by their research, we develops a large-scale model based on contrastive learning and diversity learning (Fig. 4). The model includes a feature extraction module (comprising 7 1D convolution layers), an encoder module for context modeling (consisting of 12 transformer encoding layers), and a quantization module for learning discrete common features (implemented using a linear layer). In the training process, Gumbel softmax is utilized to differentiate quantized features. Its calculation principle is as follows:

$$p_{g,v} = \frac{\exp((l_{g,v} + n_v)/\tau)}{\sum_{k=1}^{V} \exp((l_{g,k} + n_k)/\tau)}$$

Where $\tau$ is a non-negative constant, $g \in G$ , $v \in V$, $G$ is the number of codebooks, and $V$ is the dimension of each codebook.

The model achieves convergence through Contrastive Loss, which calculates the cosine similarity between context representation and the quantized representation, and Diversity Loss, which expands the space range of codebook. Their calculation principle are as follows:

$$Contrastive\ Loss = -\log \frac{\exp(sim(c_t, q_t)/\kappa)}{\sum_{\tilde{q} \sim Q_t} \exp(sim(c_t, \tilde{q})/\kappa)}$$

$c_t$ represents the output feature of the encoder at the $t$-th masked time step, $q_t$ denotes the discrete feature encoded by the quantization module at the $t$-th masked time step, and $Q_t$ includes $q_t$ and the $k$ distractors at other time steps encoded by the quantization module.

$$Diversity\ Loss = \frac{1}{GV} \sum_{g=1}^{G} -H(\bar{p}_g) = \frac{1}{GV} \sum_{g=1}^{G} \sum_{v=1}^{V} p_{g,v} \log p_{g,v}$$

Through the aforementioned pre-training process, the encoder acquires generalized features with high robustness. In the fine-tuning phase, we devised seven strategies to enhance the model's performance: 1). Adding a linear projection; 2). Incorporating an encoder layer

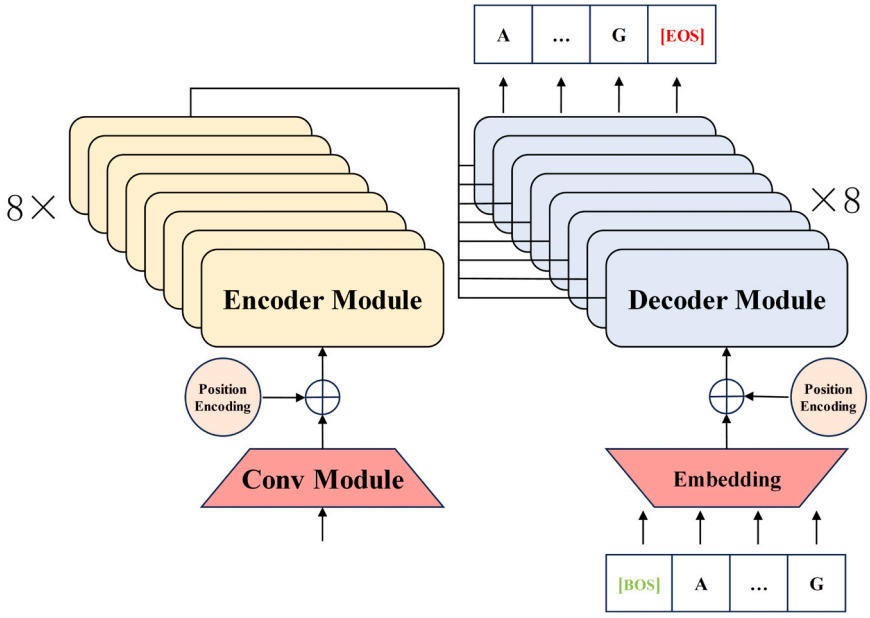


**Fig. 3.** Autoregressive Transformer-based model architecture for nanopore sequencing. The schematic illustrates the architecture of the autoregressive transformer model tailored for nanopore sequencing data. The model includes convolutional modules for feature extraction and down-sampling, an encoder composed of 8 layers for context modeling, and a decoder with 8 layers for sequence generation.

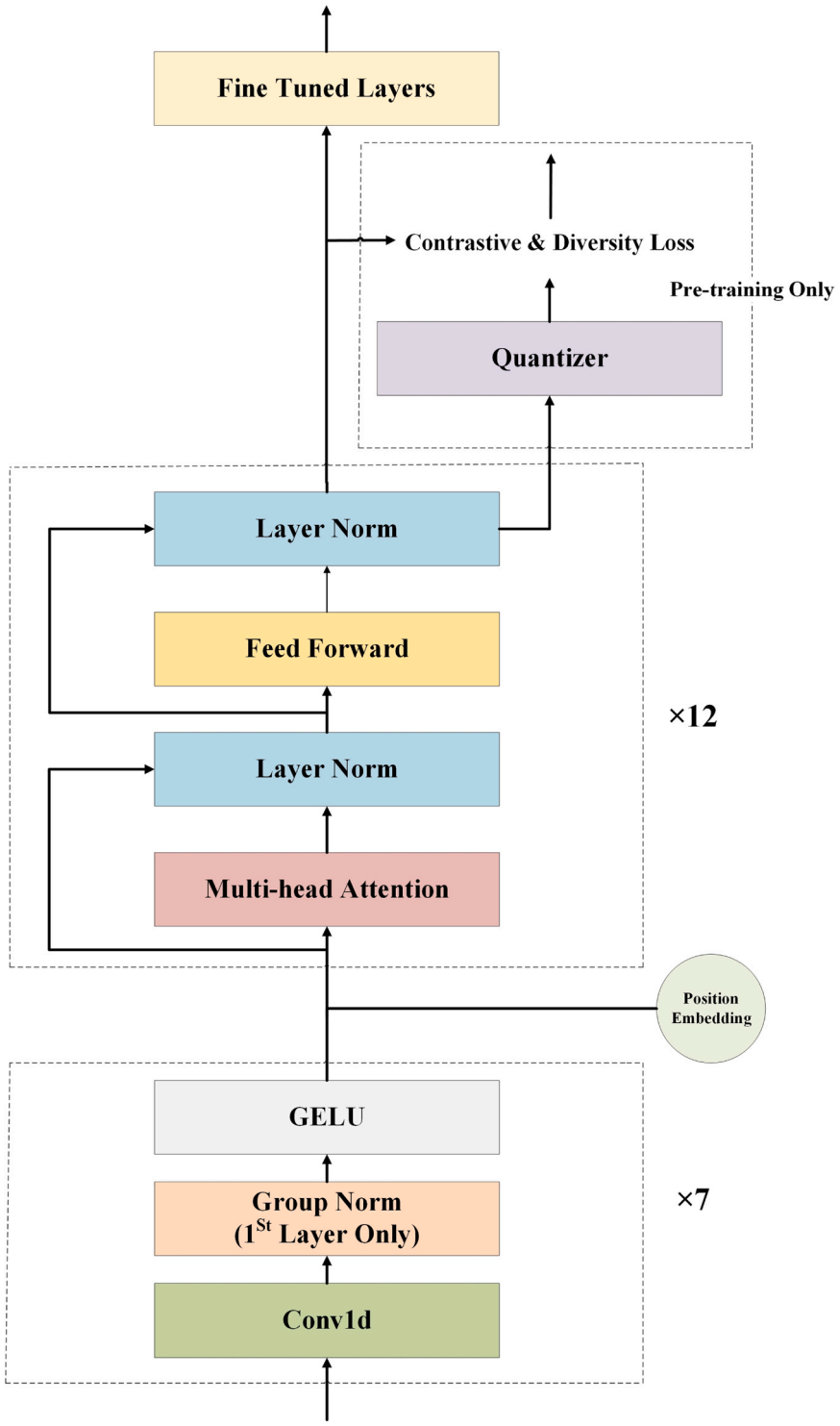


**Fig. 4.** Self-supervised large-scale model architecture in BaseNet. The schematic illustrates the self-supervised large-scale model architecture developed in BaseNet. The model comprises three key components: a feature extraction module, an encoder module for context modeling, and a quantization module for learning discrete common features through self-supervised pre-training.

followed by a linear projection; 3). Introducing two encoder layers followed by a linear projection; 4). Including three encoder layers followed by a linear projection; 5). Combining a linear layer, an encoder layer and a linear projection; 6). Including a linear layer, two encoder layers and a linear projection; and finally 7). Incorporating a linear layer, three encoder layers and a linear projection atop the models. These strategies are evaluated to determine the most suitable fine-tuning method for decoding nanopore sequencing signals. During fine-tuning, the models were optimized by minimizing the CTC loss.

### 2.5. Joint loss training and rescore mechanism

The rescore model comprises three main components: a shared encoder, a CTC decoder and attention decoder, as shown in Fig. 5. The shared encoder consists of 6 transformer encoder layers. The CTC decoder consists of a single linear layer, while the attention decoder

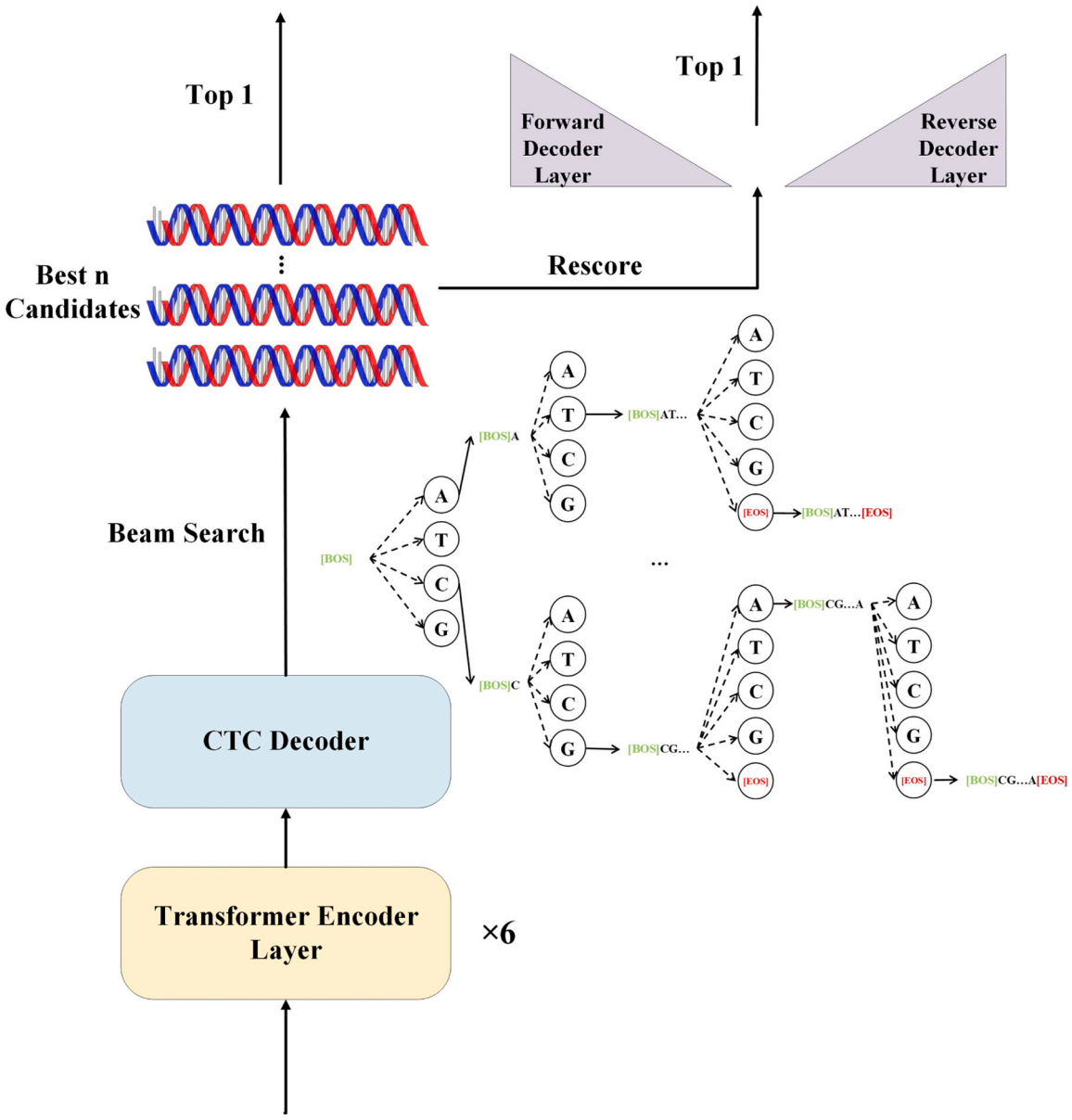


**Fig. 5.** Rescore and joint loss training model in BaseNet. This schematic illustrates the rescore and joint loss training model used in BaseNet. The model consists of three main components: a shared encoder, a CTC decoder and attention decoders. The training process utilized a joint loss to optimize performance.

consists of either three layers each of forward and reverse decoder layers (referred to as bidirectional decoder) or just three layers of forward decoder layers (referred to as unidirectional decoder) [27].

We train the model by converging a joint loss consisting of CTC loss and AED loss.

$$L_{\text{joint}}(x,y) = \lambda L_{\text{CTC}}(x,y) + (1-\lambda)L_{\text{AED}}(x,y)$$

Where $x$ is the output probability matrix, $y$ is the label, and λ is a hyperparameter between 0 and 1.

During the rescore decoding stage, the CTC decoder first utilizes CTC prefix beam search to generate $n$-best candidate sequences. Subsequently, the attention decoder rescores the candidate sequences and selects the sequence with the highest score as the final decoding output.

### 2.6. Paraformer

The Paraformer model includes an encoder for generating hidden representations, a predictor that generates acoustic embeddings and predicts sequence lengths, a sampler that randomly samples acoustic and target embeddings to create semantic embeddings, and a decoder that generates outputs based on the semantic embedding and hidden representations [28] (Fig. 6).

However, due to extensive computational requirements, each epoch of training takes over 16 h, we did not train this model because of the limitation on computational resources. Instead, the algorithm code is provided for reference and further exploration.

### 2.7. Model training

The AdamW optimizer at an initial learning rate of 0.001 and weight decay of 0.01 was used to train the models, which dynamically adjusts the learning rate to optimize the training process. BaseNet provides three learning rate schedulers: CosineDecay [33], Noam [17] and WarmupLR [34]. Fig. 7a illustrates the learning rate variation under different schedulers. Their calculation methods are as follows:

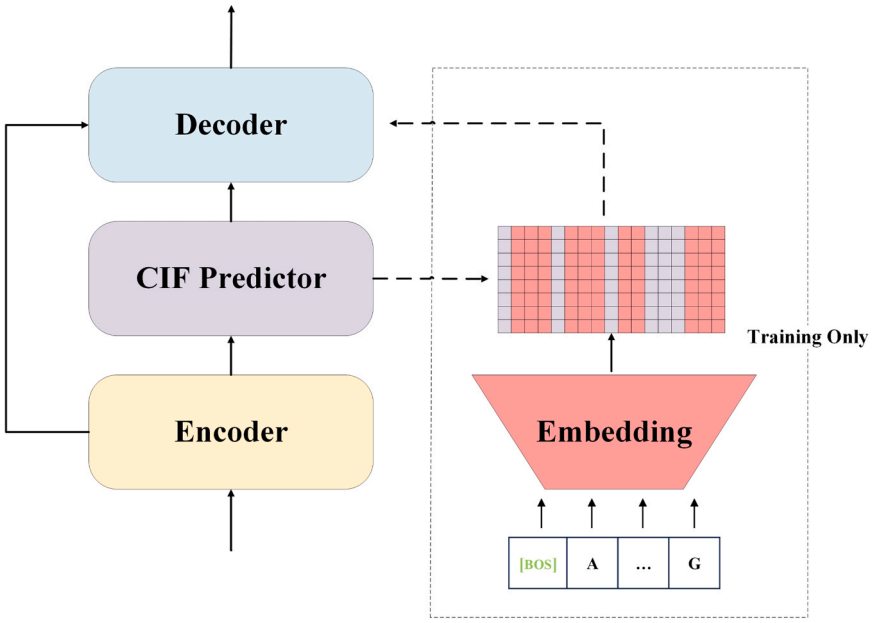


**Fig. 6.** Paraformer Architecture in BaseNet. The schematic depicts the architecture of the Paraformer model developed in BaseNet. The model features an encoder for generating hidden representations, a predictor for producing acoustic embeddings and predicting sequence lengths, a sampler for randomly creating semantic embeddings, and a decoder for generating outputs.

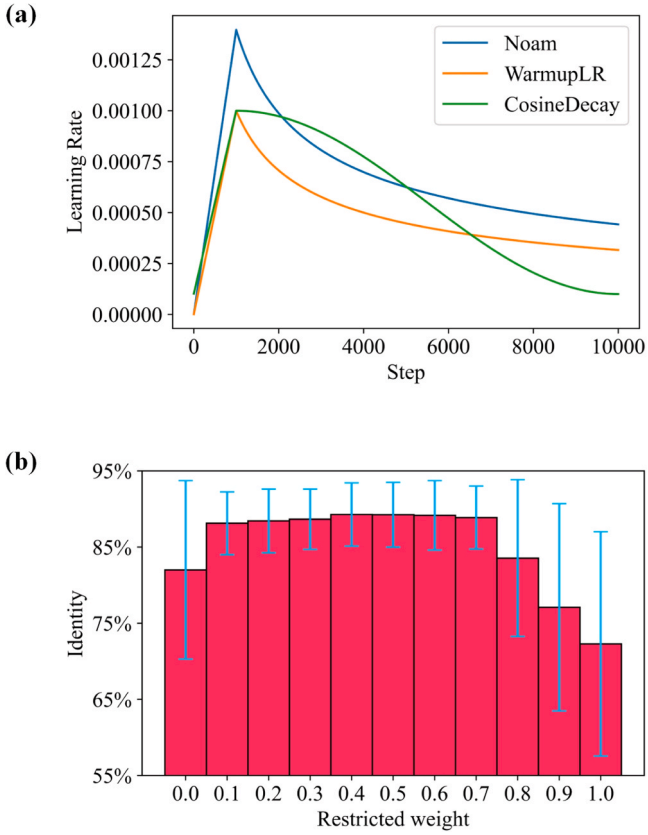


**Fig. 7.** Performance of BaseNet Under different learning rates and restricted weights. (a) Learning rate variation under three schedulers: CosineDecay, Noam, and WarmupLR. The Warmup step is set to 1000 and the total step to 10,000. (b) Autoregressive transformer prediction performance with different restricted weights. Performance significantly reduces with no EOS constraint ($w$=0) or strict constraints ($w$>=0.8). Optimal performance is observed within the range of $w$ values from 0.1 to 0.7.

$$\text{CosineDecay}: lr = \min\_val + \frac{step}{warmup\_step} \cdot (\max\_val - \min\_val) \text{ for } step \leq warmup\_step$$

$$\text{CosineDecay}: lr = \min\_val + \frac{(\max\_val - \min\_val) \cdot (1 + \cos(\pi \cdot \frac{step - warmup\_step}{total\_step - warmup\_step}))}{2} \text{ for step} > warmup\_step$$

$$\text{Noam}: lr = base\_lr \cdot model\_size^{-0.5} \cdot \min(step^{-0.5}, step \cdot warmup\_step^{-1.5})$$

$$\text{WarmupLR}: lr = base\_lr \cdot warmup\_step^{0.5} \cdot \min(step^{-0.5}, step \cdot warmup\_step^{-1.5})$$

The training was performed on 8 Nvidia A100 40 G GPUs. The autoregressive transformer (39,552,533 parameters) was trained using the cross-entropy loss function and Noam scheduler, excluding the loss for the padding token (PAD). The label sequences are smoothed with a smoothing coefficient of 0.1. The model was trained for 16 epochs with a batch size of 2. The training was performed in parallel, taking a total of 164 h. The large-scale model was pre-trained for 6 epochs with a batch size of 5, taking 209.78 h through CosineDecay scheduler. Each of the large-scale speech or signal fine-tuned and scratch models (97,137,664 parameters) was trained for 10 epochs with a batch size of 5, taking 124 h through CosineDecay scheduler. The joint loss training by WarmupLR scheduler with bidirectional decoder (29,700,511 parameters) was performed 50 epochs, consuming 240 h totally. The joint loss training by WarmupLR scheduler with unidirectional decoder (21,414,938 parameters) was performed 35 epochs, consuming 130 h totally.

### 2.8. Accuracy evaluation

The basecalling results are aligned to the reference genome using minimap2 [35] and the prediction accuracy is calculated as the similarity between the basecalled sequence and the corresponding true sequence. The similarity is defined based on the following four metrics:

Identity: the percentage of matching bases between the basecalled sequence and the true sequence. It represents the proportion of correctly identified bases.

$$Identity = \frac{Number\ of\ matched\ bases}{Length\ of\ alignment} \times 100\%$$

Mismatch rate: the percentage of bases in the basecalled sequence that do not match the true sequence. It represents the rate of incorrectly identified bases.

$$Mismatch\ rate = \frac{Number\ of\ mismatched\ bases}{Length\ of\ alignment} \times 100\%$$

Insertion rate: the percentage of bases present in the basecalled sequence but absent in the true sequence. It represents the rate of false-positive insertions.

$$Insertion\ rate = \frac{Number\ of\ inserted\ bases}{Length\ of\ alignment} \times 100\%$$

Deletion rate: the percentage of bases present in the true sequence but not detected in the basecalled sequence. It represents the rate of false-negative deletions.

$$Deletion\ rate = \frac{Number\ of\ deleted\ bases}{Length\ of\ alignment} \times 100\%$$

The overall median values of the above metrics were used to compare our model with other methods, which were also adopted in multiple basecaller studies for performance evaluation and comparation.

## 3. Results and discussion

### 3.1. Autoregressive termination constraint

In autoregressive decoding, it is crucial to appropriately constrain the generation of the end of sequence (EOS) token to achieve optimal model performance. Here, we introduce a parameter called the constraint weight ($w$). The value of $w$ ranges from 0 to 1. $w$= 0 indicates no constraint on the generation of the EOS token, and $w$ = 1 means that the model does not consider generating the EOS token and decode until it reaches the maximum length. We compared the prediction accuracy of autoregressive transformer under different values of $w$ on the same test data (Fig. 7b). The results indicate that the model performance significantly deteriorates when there is no constraint on EOS generation ($w$=0) or when there is a strict constraint ($w$>=0.8). The model performs stably and well within the range of 0.1 to 0.7.

### 3.2. Mapping between current signal and base sequence in the cross-attention layer

The cross-attention mechanism enables the decoder to derive hidden representations of the electrical signals from the encoder's output and apply them to the base sequence decoding process. In the cross-attention layer, the queries $Q$ come from the base sequences, which serve as the input to the decoder, while the keys $K$ and values $V$ are derived from the encoder's output, originating from the current signals. This mechanism establishes attention relationships between the generated base sequences and the current signals.

At each position in Q, the cross-attention mechanism calculates the attention weights between the current base and all positions in the encoder's output. This allows the decoder to focus on and utilize the corresponding current waveform information related to the base sequence. In the temporal domain, there is theoretically a linear correspondence between the current signals and the base sequences. Specifically, changes in the base sequences result in corresponding changes in current signals, and these changes are approximately linear.

Next we investigate whether transformer models can learn and capture this correlated relationship. To achieve this, attention weights were extracted and visualized from the cross-attention layers in autoregressive transformer (Fig. 8 and S1). It is revealed that the strength of the linear relationship varies across different layers of the transformer decoder. Specifically, the relationship exhibits a weak-strong-weak pattern, with the highest strength observed in the fourth to sixth layers, indicating these layers are most effective at capturing the linear correlation. Conversely, the linear relationship is weaker or absent in the first to third layers and the seventh to eighth layers.

The lack or weakening of the linear relationship in the first to third layers can be attributed to their focus on local features and limited ability to capture the overall linear relationship. These layers may primarily concentrate on extracting local patterns and features, resulting in

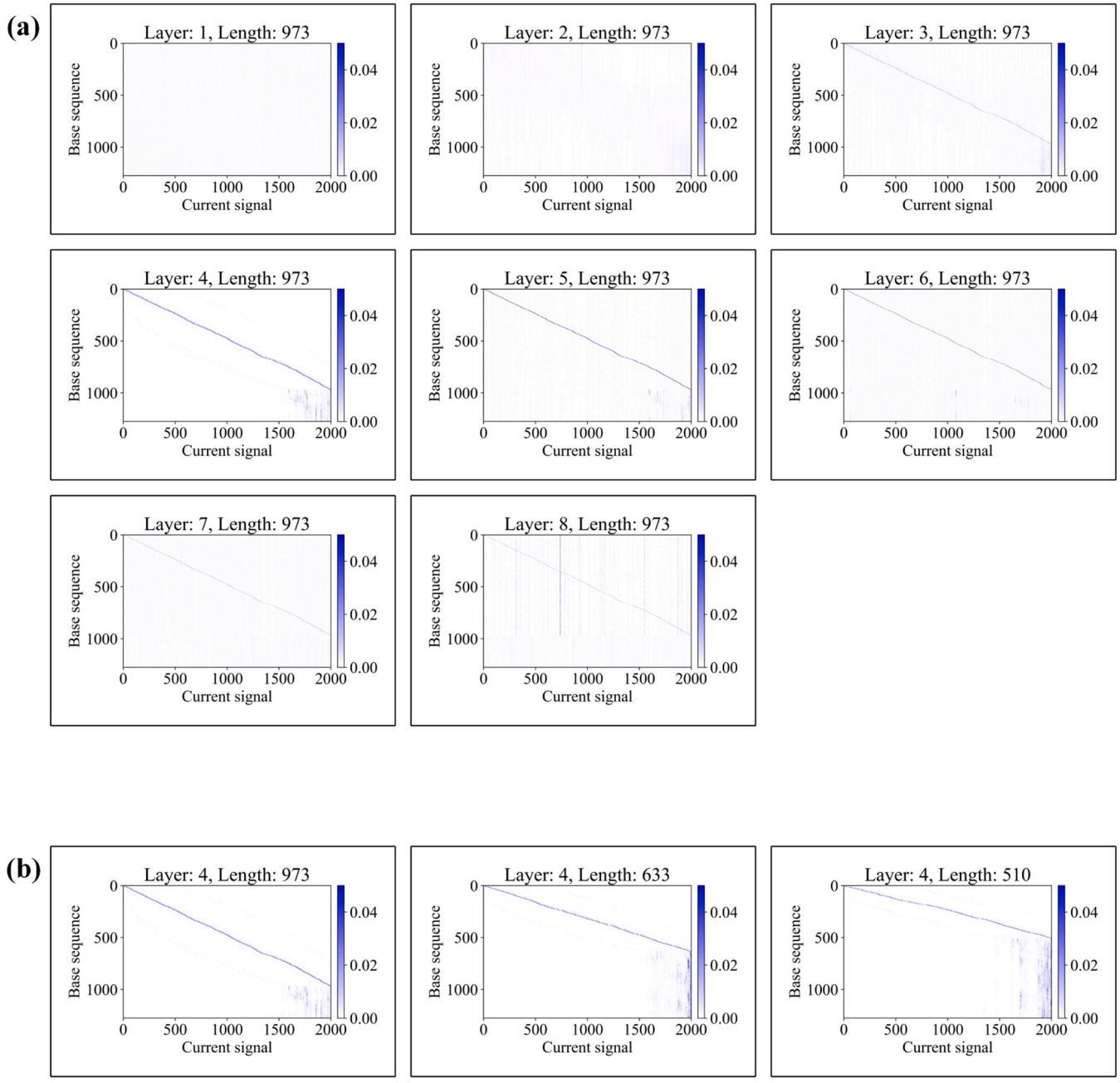





**Fig. 8.** Cross-attention weights between current signals and base sequences in transformer decoder. (a) Visualization of cross-attention weights across different decoder layers for a specific sequence. The linear relationship between signal and sequence follows a weak-strong-weak pattern among layers. Layers four to six exhibit the strongest linear relationships, indicating that they effectively capture both local and global features. In contrast, layers one to three and seven to eight show no or weaker linear relationships, indicating a focus on either local or global features without integrating both. (b) Visualization of cross-attention weights across different sequences in decoder layer four.

a weaker capacity to learn the global linear relationship. The weakening of the linear relationship in the seventh and eighth layers can be attributed to their emphasis on global features while overlooking the local linear relationship. Both layers may excel in capturing global patterns and features but exhibit a weaker ability to learn the local linear relationships.

In contrast, the fourth to sixth layers demonstrate the strongest linear relationship, suggesting that they strike a better balance in attention weight learning, allowing them to simultaneously capture both local and global features. These layers appear to be more proficient in capturing the linear relationship between the base sequences and the current signals. It is particularly noteworthy that in the fourth layer, the attention heads all exhibit a strong linear relationship.

Thus, it is suggested that the attention weights from the fourth to sixth of the cross-attention layers can be extracted to establish an appropriate weight threshold. To determine this threshold, one potential approach involves analyzing the distribution of attention weights across the fourth to sixth layers of the cross-attention mechanism. By examining this distribution, researchers could identify natural cutoffs or peaks that may serve as effective thresholds. Another method could involve the application of statistical techniques, such as selecting a threshold based on the percentile of weights, to isolate the most significant attention contributions. By doing so, the current signal points that exhibit a strong correlation with specific bass can be identified. This

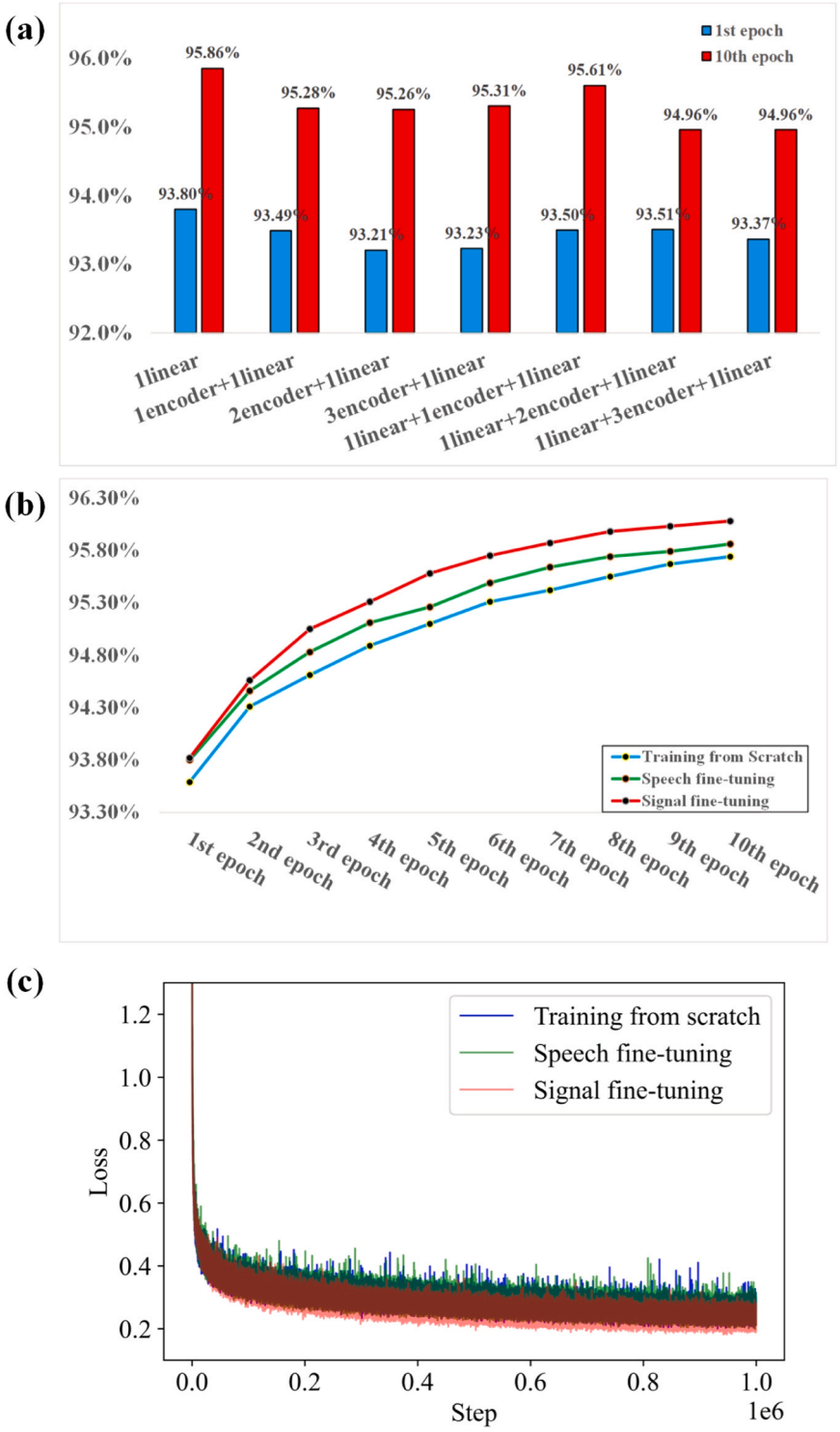


**Fig. 9.** Performance comparison of different large-scale models. (a) Performance comparison of different fine-tuned models. Among the 7 fine-tuned models, the model with one additional linear layer achieved the best performance, reaching 95.86 % identity after 10 epochs of fine-tuning. (b) Accuracy comparison of large models under different training conditions as indicated. (c) Training loss curves for signal fine-tuning, speech fine-tuning, and training from scratch.

approach enables the assignment of current signal points to their corresponding bases, thereby establishing the correspondence between current signal points and bases to achieve alignment. This alignment method, based on attention weights and thresholds, offers a new strategy for data chunking and training set construction.

### 3.3. *Performance of large-scale model pre-training and fine-tuning*

In this study, the large-scale model underwent five rounds of pre-training based on nanopore sequencing signals, followed by fine-tuning (signal fine-tuning). To determine the optimal fine-tuning strategy, we loaded the pre-training weights of wav2vec2.0 into our large-scale model and performed fine-tuning training (speech fine-tuning). After 10 epochs, we found that the model with only one added linear

layer exhibited the best decoding performance, achieving 95.86 % identity (Fig. 9a).

To address concerns regarding potential overfitting, we carefully monitored corresponding metrics throughout the training process (Fig. S2). Specifically, we tracked the loss on the validation set and the training set across each epoch, as well as monitoring the identity on the validation set. The results demonstrated that the loss on the validation set consistently decreased alongside the training set loss, without any divergence that would indicate overfitting. Furthermore, the validation identity improved steadily throughout the training process. These observations confirm that all 7 fine-tuned models did not experience overfitting, maintaining strong generalization capabilities throughout the training phase.

The finding indicates that incorporating too many or overly complex fine-tuning layers in large-scale pre-trained models does not necessarily enhance performance. Instead, adding just a linear layer based on the requirements of the task can yield desirable results. Excessive or complex fine-tuning layers may introduce too many parameters and complexity, leading to overfitting or performance degradation. Therefore, in subsequent fine-tuning studies, we opted to add only one linear layer.

Simultaneously, we also trained the large model from scratch using CTC loss (training from scratch). The results (Fig. 9b-c and Supplemental Table 1) indicate that signal fine-tuning yields the best performance. This is understandable, as during the pre-training process, the large-scale model learns efficient, robust, and generic high-dimensional features from the input signals. Fine-tuning with supervised data refines these 'generic features' into 'task-specific features', thereby effectively applying these features to downstream tasks. Notably, the evaluation accuracy of speech fine-tuning significantly surpasses that of training from scratch, despite having similar training curves.

In speech recognition, the model extracts speech features from the

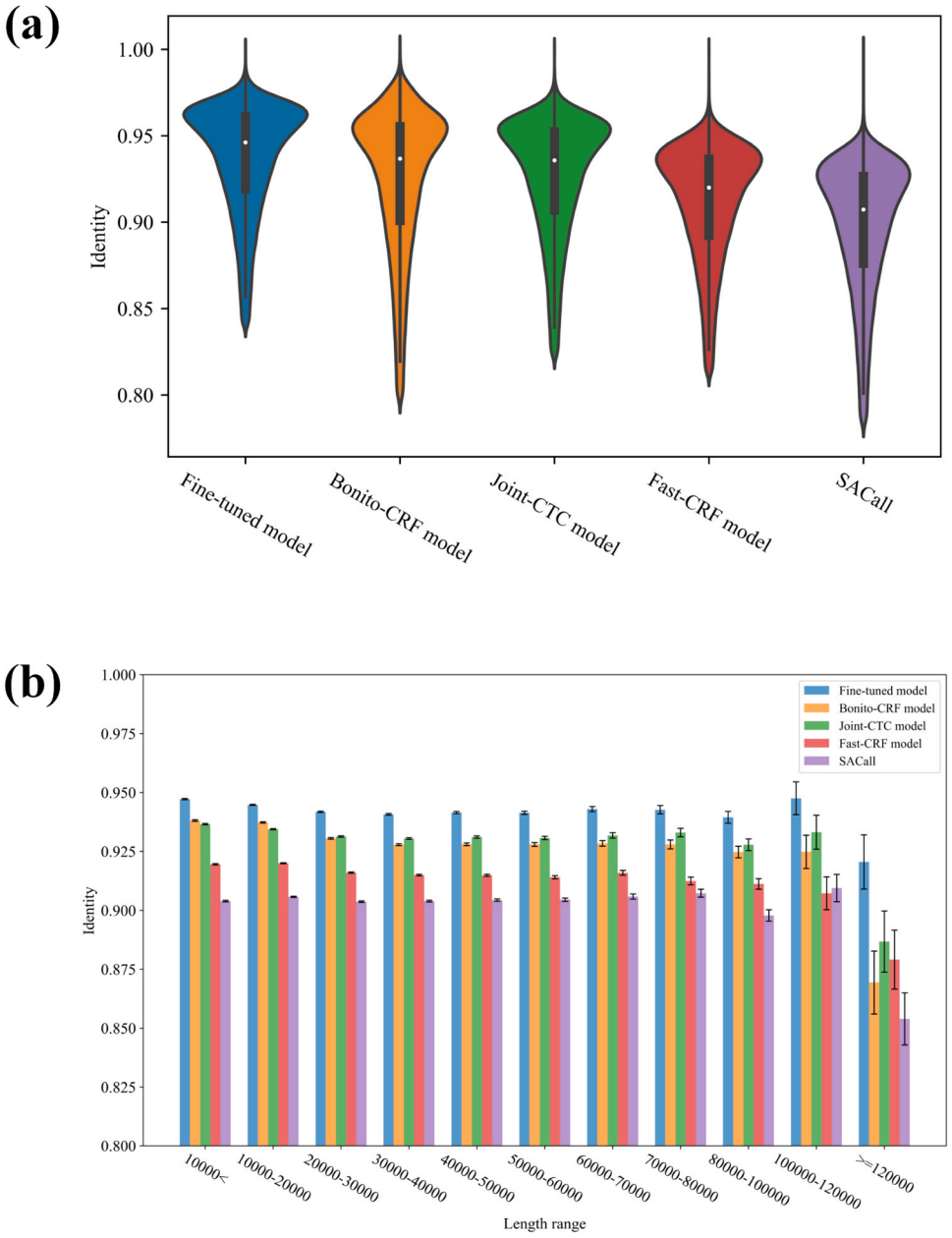


**Fig. 10.** Performance comparison of different basecallers. (a) Identity comparison of different basecallers. (b) Identity comparison of different basecallers under different sequence lengths. Error bars represent standard errors. Fine-tuned model performs best across all lengths, and when the sequence length exceeds 20,000, the Joint-CTC model outperforms the Bonito-CRF model.

original audio signal, while in nanopore sequencing current decoding, it extracts base sequence information from the current signal. These two tasks share similarities in signal extraction processes. Additionally, both speech recognition and nanopore sequencing current decoding require the model to recognize and learn specific patterns in the data.

These findings suggest shared underlying structures or patterns between speech recognition and nanopore sequencing current decoding. The high-dimensional hidden representations for both tasks exhibit similarities within the model, indicating that current signals and speech waveforms share common 'generic features'.

When researchers discover a new nanopore protein, the task of rapidly and cost-effectively constructing a high-quality dataset from scratch can be overwhelming. Insights from above studies suggest a potential solution to this challenge. By performing unsupervised pre-training on large-scale models, researchers can leverage existing knowledge and subsequently apply few-shot learning using a limited set of known full-length sequencing signals. This approach can reduce the need for extensive labeled data and expedite the initial development of decoding models, enabling quicker iterations toward the creation of high-quality datasets and the refinement of high-precision models. In future studies, we will experiment with this approach, assessing the effectiveness of combining large-scale pre-training with few-shot to rapidly adapt BaseNet to new nanopore protein.

### 3.4. Joint loss training improves model performance

Next, we compared the performance of unidirectional and bidirectional decoder rescore models in the task of nanopore sequencing signal decoding. The models were evaluated at the 35th epoch, with the unidirectional decoder achieving an identity of 94.57 %, while the bidirectional decoder slightly outperformed it at 94.66 % (Supplemental Table 2). Both models were trained using a combination of weighted CTC and AED loss (joint loss), enabling independent decoding by the encoder.

To further investigate the decoding capabilities of the encoder, we conducted ablation experiments by comparing the performance of the encoder-only approach with the encoder-decoder architecture. Surprisingly, both the unidirectional and bidirectional decoders achieved comparable performance to their respective complete models when using the encoder-only approach (Supplemental Table 2).

Subsequently, we focused on training the encoder alone using the CTC loss. The results revealed that training the encoder solely with CTC loss resulted in 94.18 % identity after 35th epoch (Supplemental Table 2).

These findings suggest that the non-autoregressive transformer rescore mechanism has limited potential for performance improvement in nanopore sequencing signal decoding. However, joint loss training of the model using CTC and attention mechanisms proves to be highly effective in enhancing model performance.

### 3.5. Comparison of different basecallers

We compared the performance of five different basecallers, which are deep learning-based decoding models for sequencing data, in terms of decoding accuracy and inference speed. These basecallers include the Fine-tuned model, which is a large-scale pre-trained model fine-tuned with one linear layer, the Joint-CTC model trained through joint loss and bidirectional decoder, the Bonito's CRF model, the Fast-CRF model which replaces LSTM layers of Bonito-CRF model with fast attention layers, and the latest third party open source basecaller SACall.

The inference was conducted on an NVIDIA RTX 3090 24 G GPU. The Fine-tuned model outperformed the latest ONT basecaller in decoding accuracy after only 10 epochs of training (independent t-test, $p = 0.0$; Fig. 10a, S3a,S3b, and S3c, and Supplemental Table 3–4). Although the performance of Joint-CTC model and the Fast-CRF model lag behind the Bonito-CRF model (independent t-test, Joint-CTC: $p = 3.67e\text{-}57$, Fast-CRF: $p = 0.0$), they significantly surpass SACall (independent t-test, Joint-CTC: $p = 0.0$, Fast-CRF: $p = 0.0$). Further, the performance of various basecallers was compared at different decoding lengths (Fig. 10b, S3d, S3e and S3f). It can be seen that the Fine-tuned model performs best across all lengths, and when the sequence length exceeds 20,000, the Joint-CTC model outperforms the Bonito-CRF model. These findings establish that BaseNet achieves reasonable performance and compares favorably with the ONT basecaller. Thus, BaseNet provides the state-of-the-art open source basecallers.

We acknowledge that the current decoding speed of BaseNet, while providing competitive accuracy, is slower compared to existing ONT basecallers. This is primarily due to the complexity inherent in the Transformer-based models that BaseNet employs, as well as its implementation in Python, which prioritizes flexibility and accessibility for researchers. In contrast, ONT basecallers benefit from highly optimized production code written in C++ and CUDA. To address this, we are actively pursuing several optimization strategies. Our future work will focus on techniques such as model pruning, knowledge distillation, and quantization, which have the potential to significantly reduce computational demands while maintaining accuracy. Additionally, we are exploring more efficient transformer architectures, such as Linformer [36] and Longformer [37], to further enhance BaseNet's speed. We are also considering the integration of hardware-aware techniques like Flash-attention [38,39] to improve performance. These optimization efforts are a priority in our ongoing and future research.

While BaseNet was originally developed for decoding DNA sequencing signals, its model architecture is theoretically adaptable for RNA basecalling. However, due to the distinct physical and chemical properties of RNA, such as differences in pore velocity and direction compared to DNA, specific adjustments to the model are necessary. These adjustments may include increasing the convolutional stride, increasing the model complexity, and implementing reverse decoding strategies. In future work, we plan to explore these modifications and assess BaseNet's performance in RNA sequencing applications.

## 4. Conclusion

We introduce BaseNet, an open-source toolkit for nanopore sequencing signal decoding based on state-of-the-art transformer algorithm. Experiments and comparisons between BaseNet and other basecallers demonstrate that BaseNet achieves reasonable performance and comparable results. In addition, our study reveals several insights: cross-attention weights of transformer effectively map the correspondence between current signals and base sequences; joint loss training with the addition of forward and reverse decoders aids in better model convergence; large-scale pre-trained model achieve higher decoding accuracy; and there are common 'generic features' between speech waveforms and sequencing signals in model representation.

## Ethical considerations

All ethical standards and guidelines applicable to the research, including data usage and analysis, have been followed.

## Authorship consent

All authors have read and approved the final version of the manuscript for submission and agree with the order of authorship as listed.

1. **Original Work**:
2. The manuscript is our original work and has not been submitted or published elsewhere, either in part or in whole.
3. **Contribution and Responsibility**:
4. All authors have significantly contributed to the research, analysis, writing, and review of the manuscript. We take full responsibility for

the content presented and affirm that the data, interpretations, and conclusions are based on valid scientific methods.

## Acknowledgment of feedback

We have carefully considered and addressed all feedback from the reviewers in our revised manuscript, and we believe the changes made are in line with the journal's guidelines and expectations. We sincerely appreciate the opportunity to submit this revised version of our manuscript and look forward to the editorial review process.

## CRediT authorship contribution statement

**Jizhong Lou:** Writing – review & editing. **Chen Sun:** Writing – review & editing. **Daqian Wang:** Writing – review & editing. **Qingwen Li:** Writing – original draft, Validation, Software, Methodology, Investigation.

## Declaration of Competing Interest

Daqian Wang and Jizhong Lou are co-founders and shareholders of Beijing Polyseq Biotech Co. Ltd. Beijing Polyseq Biotech Co. Ltd. and Institute of Biophysics, Chinese Academy of Sciences have filed a patent using materials described in this article.

## Availability and Implementation

The data of this study was taken from the study of Wick et. al (https://doi.org/10.1099/mgen.0.000132.).

The code, model weights of the Pre-trained and Fine-tuned large-scale model, Joint-CTC model, and Fast-CRF model to BaseNet can be accessed from Github (https://github.com/liqingwen98/BaseNet).

## Acknowledgements

This work is partly supported by grants from the Ministry of Science and Technology of China (2019YFA0707001 and 2021YFF0700201) and the Strategic Priority Research Program of the Chinese Academy of Sciences (XDB37020102).

## Appendix A. Supporting information

Supplementary data associated with this article can be found in the online version at doi:10.1016/j.csbj.2024.09.016.

## References

[1] Jain M, Koren S, Miga KH, Quick J, Rand AC, Sasani TA, et al. Nanopore sequencing and assembly of a human genome with ultra-long reads. Nat Biotechnol 2018;36(4):338–45. https://doi.org/10.1038/nbt.4060.
[2] Davenport CF, Scheithauer T, Dunst A, Bahr FS, Dorda M, Wiehlmann L, et al. Genome-Wide Methylation Mapping Using Nanopore Sequencing Technology Identifies Novel Tumor Suppressor Genes in Hepatocellular Carcinoma. Int J Mol Sci 2021;22(8):3937. https://www.mdpi.com/1422-0067/22/8/3937.
[3] Quick J, Loman NJ, Duraffour S, Simpson JT, Severi E, Cowley L, et al. Real-time, portable genome sequencing for Ebola surveillance. Nature 2016;530(7589):228–32. https://doi.org/10.1038/nature16996.
[4] Wang J, Moore NE, Deng YM, Eccles DA, Hall RJ. MinION nanopore sequencing of an influenza genome. Front Microbiol 2015;6:766. https://doi.org/10.3389/fmicb.2015.00766.
[5] Faria NR, Quick J, Claro IM, Thézé J, de Jesus JG, Giovanetti M, et al. Establishment and cryptic transmission of Zika virus in Brazil and the Americas. Nature 2017;546(7658):406–10. https://doi.org/10.1038/nature22401.
[6] Yakovleva A, Kovalenko G, Redlinger M, Liulchuk MG, Bortz E, Zadorozhna VI, et al. Tracking SARS-COV-2 variants using Nanopore sequencing in Ukraine in 2021. Sci Rep 2022;12(1):15749. https://doi.org/10.1038/s41598-022-19414-y.
[7] Boykin LM, Sseruwagi P, Alicai T, Ateka E, Mohammed IU, Stanton J-AL, et al. Tree lab: portable genomics for early detection of plant viruses and pests in sub-Saharan Africa. Genes 2019;10(9):632. https://www.mdpi.com/2073-4425/10/9/632.
[8] De Coster W, D'Hert S, Schultz DT, Cruts M, Van Broeckhoven C. NanoPack: visualizing and processing long-read sequencing data. Bioinformatics 2018;34(15):2666–9. https://doi.org/10.1093/bioinformatics/bty149.
[9] Lin B, Hui J, Mao H. Nanopore Technology and Its Applications in Gene Sequencing. Biosens (Basel) 2021;11(7). https://doi.org/10.3390/bios11070214.
[10] Amodei, D., Anubhai, R., Battenberg, E., Case, C., Casper, J., Catanzaro, B., et al. (2015). Deep Speech 2: End-to-End Speech Recognitionin English and Mandarin. arXiv:1512.02595. Retrieved December 01, 2015, from https://ui.adsabs.harvard.edu/abs/2015arXiv151202595A.
[11] Bai, S., Zico Kolter, J., & Koltun, V. (2018). An Empirical Evaluation of Generic Convolutional and Recurrent Networks for Sequence Modeling. arXiv:1803.01271. Retrieved March 01, 2018, from https://ui.adsabs.harvard.edu/abs/2018arXiv180301271B.
[12] David M, Dursi LJ, Yao D, Boutros PC, Simpson JT. Nanocall: an open source basecaller for Oxford Nanopore sequencing data. Bioinformatics 2016;33(1):49–55. https://doi.org/10.1093/bioinformatics/btw569.
[13] Stoiber M, Brown J. BasecRAWller: streaming nanopore basecalling directly from raw signal. bioRxiv 2017:133058. https://doi.org/10.1101/133058Technologies.
[14] Teng H, Cao MD, Hall MB, Duarte T, Wang S, Coin LJ. Chiron: translating nanopore raw signal directly into nucleotide sequence using deep learning. GigaScience 2018;7(5):giy037.
[15] Zeng J, Cai H, Peng H, Wang H, Zhang Y, Akutsu T. Causalcall: Nanopore basecalling using a temporal convolutional network. Front Genet 2020:1332.
[16] Boža V, Brejová B, Vinař T. DeepNano: Deep recurrent neural networks for base calling in MinION nanopore reads. PLoS One 2017;12(6):e0178751. https://doi.org/10.1371/journal.pone.0178751.
[17] Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., et al. (2017). Attention Is All You Need. arXiv:1706.03762. Retrieved June 01, 2017, from https://ui.adsabs.harvard.edu/abs/2017arXiv170603762V.
[18] Carion, N., Massa, F., Synnaeve, G., Usunier, N., Kirillov, A., & Zagoruyko, S. (2020). End-to-End Object Detection with Transformers. arXiv:2005.12872. Retrieved May 01, 2020, from https://ui.adsabs.harvard.edu/abs/2020arXiv200512872C.
[19] Dosovitskiy, A., Beyer, L., Kolesnikov, A., Weissenborn, D., Zhai, X., Unterthiner, T., et al. (2020). An Image is Worth 16x16 Words: Transformers for Image Recognition at Scale. arXiv:2010.11929. Retrieved October 01, 2020, from https://ui.adsabs.harvard.edu/abs/2020arXiv201011929D.
[20] Zheng, S., Lu, J., Zhao, H., Zhu, X., Luo, Z., Wang, Y., et al. (2020). Rethinking Semantic Segmentation from a Sequence-to-Sequence Perspective with Transformers. arXiv:2012.15840. Retrieved December 01, 2020, from https://ui.adsabs.harvard.edu/abs/2020arXiv201215840Z.
[21] Radford, A., Kim, J. W., Xu, T., Brockman, G., McLeavey, C., & Sutskever, I. (2022). Robust Speech Recognition via Large-Scale Weak Supervision. arXiv:2212.04356. Retrieved December 01, 2022, from https://ui.adsabs.harvard.edu/abs/2022arXiv221204356R.
[22] Schneider, S., Baevski, A., Collobert, R., & Auli, M. (2019). wav2vec: Unsupervised Pre-training for Speech Recognition. arXiv:1904.05862. Retrieved April 01, 2019, from https://ui.adsabs.harvard.edu/abs/2019arXiv190405862S.
[23] Ferruz N, Schmidt S, Höcker B. ProtGPT2 is a deep unsupervised language model for protein design. Nat Commun 2022;13(1):4348. https://doi.org/10.1038/s41467-022-32007-7.
[24] Madani A, Krause B, Greene ER, Subramanian S, Mohr BP, Holton JM, et al. Large language models generate functional protein sequences across diverse families. Nat Biotechnol 2023. https://doi.org/10.1038/s41587-022-01618-2.
[25] Wang R, Jiang Y, Jin J, Yin C, Yu H, Wang F, et al. DeepBIO: an automated and interpretable deep-learning platform for high-throughput biological sequence prediction, functional annotation and visualization analysis. Nucleic Acids Res 2023;51. https://doi.org/10.1093/nar/gkad055.
[26] Huang N, Nie F, Ni P, Luo F, Wang J. SACall: A Neural Network Basecaller for Oxford Nanopore Sequencing Data Based on Self-Attention Mechanism. IEEE/ACM Trans Comput Biol Bioinform 2022;19(1):614–23. https://doi.org/10.1109/tcbb.2020.3039244.
[27] Deng, K., Cao, S., Zhang, Y., & Ma, L. (2021). Improving Hybrid CTC/Attention End-to-end Speech Recognition with Pretrained Acoustic and Language Model. arXiv:2112.07254. Retrieved December 01, 2021, from https://ui.adsabs.harvard.edu/abs/2021arXiv211207254D.
[28] Gao, Z., Zhang, S., McLoughlin, I., & Yan, Z. (2022). Paraformer: Fast and Accurate Parallel Transformer for Non-autoregressive End-to-End Speech Recognition. arXiv:2206.08317. Retrieved June 01, 2022, from https://ui.adsabs.harvard.edu/abs/2022arXiv220608317G.
[29] Baevski, A., Zhou, H., Mohamed, A., & Auli, M. (2020). wav2vec 2.0: A Framework for Self-Supervised Learning of Speech Representations. arXiv:2006.11477. Retrieved June 01, 2020, from https://ui.adsabs.harvard.edu/abs/2020arXiv200611477B.
[30] Wu, C., Wu, F., Qi, T., Huang, Y., & Xie, X. (2021). Fastformer: Additive Attention Can Be All You Need. arXiv:2108.09084. Retrieved August 01, 2021, from https://ui.adsabs.harvard.edu/abs/2021arXiv210809084W.
[31] Wick RR, Judd LM, Holt KE. Performance of neural network basecalling tools for Oxford Nanopore sequencing. Genome Biol 2019;20(1):129. https://doi.org/10.1186/s13059-019-1727-y.
[32] Technologies, O. N. Guppy. https://community.nanoporetech.com/downloads.
[33] Loshchilov, I., & Hutter, F. (2016). SGDR: Stochastic Gradient Descent with Warm Restarts. arXiv:1608.03983. Retrieved August 01, 2016, from https://ui.adsabs.harvard.edu/abs/2016arXiv160803983L.
[34] Gao, Z., Li, Z., Wang, J., Luo, H., Shi, X., Chen, M., et al. (2023). FunASR: A Fundamental End-to-End Speech Recognition Toolkit. arXiv:2305.11013. Retrieved May 01, 2023, from https://ui.adsabs.harvard.edu/abs/2023arXiv230511013G.

[35] Li H. Minimap2: pairwise alignment for nucleotide sequences. Bioinformatics 2018;34(18):3094–100. https://doi.org/10.1093/bioinformatics/bty191.
[36] Wang, S., Li, B. Z., Khabsa, M., Fang, H., & Ma, H. (2020). Linformer: Self-Attention with Linear Complexity. arXiv:2006.04768. Retrieved June 01, 2020, from https://ui.adsabs.harvard.edu/abs/2020arXiv200604768W.
[37] Beltagy, I., Peters, M. E., & Cohan, A. (2020). Longformer: The Long-Document Transformer. arXiv:2004.05150. Retrieved April 01, 2020, from https://ui.adsabs.harvard.edu/abs/2020arXiv200405150B.
[38] Dao, T., Fu, D. Y., Ermon, S., Rudra, A., & Ré, C. (2022). FlashAttention: Fast and Memory-Efficient Exact Attention with IO-Awareness. arXiv:2205.14135. Retrieved May 01, 2022, from https://ui.adsabs.harvard.edu/abs/2022arXiv220514135D.
[39] Dao, T. (2023). FlashAttention-2: Faster Attention with Better Parallelism and Work Partitioning. arXiv:2307.08691. Retrieved July 01, 2023, from https://ui.adsabs.harvard.edu/abs/2023arXiv230708691D.