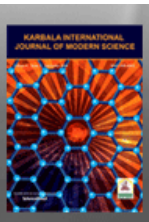

Karbala International Journal of Modern Science

Volume 12 | Issue 2 Article 8

Inc3ViTs Model: A Hybrid Architecture to Accelerate and Reduce Complexity for the DeepVariant Model for Variant Calling

Mustafa Al-Saffar
Software Dept., Information Technology College, University of Babylon, Hilla, Iraq,,
mustafaalsaffar.sw.msc@student.uobabylon.edu.iq

Sura Z. Al Rashid
Software Dept., Information Technology College, University of Babylon, Hilla, Iraq,

Inc3ViTs Model: A Hybrid Architecture to Accelerate and Reduce Complexity for the DeepVariant Model for Variant Calling

Abstract

Deep learning has revolutionized genomic variant calling, yet the computational cost of current systems continues to limit scalability. We present a controlled efficiency study of DeepVariant-style pileup architectures under identical training and inference conditions, comparing architectural downsizing with a hybrid CNN–local attention design. Inc3ViTs pairs a streamlined InceptionV3 stem with a lightweight local attention head based on patch tokenization and windowed self-attention, enabling a direct comparison with a CNN-only reduced Inception baseline. Across whole-genome and whole-exome short-read datasets, Inc3ViTs reduces training time by ~40–50% and reduces inference runtime relative to the original DeepVariant. The CNN-only baseline indicates that most speedups stem from architectural simplification, whereas the hybrid design achieves competitive accuracy with statistically comparable SNP performance and marginal INDEL trade-offs at high coverage, alongside improved robustness at lower sequencing depth. Our evaluation is restricted to GIAB short-read small-variant benchmarking (SNPs and INDELs); structural variants and long-read settings were not evaluated. Experiments were conducted on a single workstation equipped with an RTX 3070 Ti under controlled hardware conditions.

Keywords

Variant calling; Deep Learning; Vision Transformers; Local attention; Genomic analysis; Short-read sequencing; DeepVariant

Cover Page Footnote

Funding: No external funding was received for this study. Conflict of Interest: The authors declare no conflict of interest.

RESEARCH PAPER

Inc3ViTs Model: A Hybrid Architecture to Accelerate and Reduce Complexity for the DeepVariant Model for Variant Calling

Mustafa Al-Saffar*, Sura Z. Al Rashid

Software Dept., Information Technology College, University of Babylon, Hilla, Iraq

Abstract

Deep learning has revolutionized genomic variant calling, yet the computational cost of current systems continues to limit scalability. We present a controlled efficiency study of DeepVariant-style pileup architectures under identical training and inference conditions, comparing architectural downsizing with a hybrid CNN−local attention design. Inc3ViTs pairs a streamlined InceptionV3 stem with a lightweight local attention head based on patch tokenization and windowed self-attention, enabling a direct comparison with a CNN-only reduced Inception baseline. Across whole-genome and whole-exome short-read datasets, Inc3ViTs reduces training time by ~40−50% and reduces inference runtime relative to the original DeepVariant. The CNN-only baseline indicates that most speedups stem from architectural simplification, whereas the hybrid design achieves competitive accuracy with statistically comparable SNP performance and marginal INDEL trade-offs at high coverage, alongside improved robustness at lower sequencing depth. Our evaluation is restricted to GIAB short-read small-variant benchmarking (SNPs and INDELs); structural variants and long-read settings were not evaluated. Experiments were conducted on a single workstation equipped with an RTX 3070 Ti under controlled hardware conditions.

Keywords: Variant calling, Deep learning, Vision transformers, Local attention, Genomic analysis, Short-read sequencing, DeepVariant

1. Introduction

With the advent of next-generation sequencing, genome sequencing has become increasingly essential [1,2]. As technical limitations have been overcome [3,4], bioinformatics methods have matured and now support clinically actionable genomic interpretation and therapeutic decision-making [3,5]. Genome sequencing and variant detection underpin many applications, including identifying disease-causing variants and informing therapies that target disease-related genes [5]. Google's DeepVariant is a pioneering deep learning approach that encodes aligned reads as a multi-channel pileup image and performs variant calling as a multiclass classification task [6].

DeepVariant has been evaluated in multiple benchmarking studies and has been reported to achieve high accuracy [6−8]. It performs strongly on short-read SNP and small INDEL calling in GIAB-based evaluations [6,8]. However, DeepVariant relies on an InceptionV3 CNN backbone in the call_variants stage, which contributes to substantial computational cost [9,10]. Training can be time-consuming and resource-intensive, which can complicate deployment and transfer learning in compute-constrained environments [9].

Most practical improvements to DeepVariant have focused on expanding training datasets and

Received 19 December 2025; revised 15 March 2026; accepted 19 March 2026.
Available online 20 April 2026

* Corresponding author.
E-mail addresses: mustafaalsaffar.sw.msc@student.uobabylon.edu.iq (M. Al-Saffar), sura.alrashid@uobabylon.edu.iq (S.Z. Al Rashid).

https://doi.org/10.33640/2405-609X.3460

refining preprocessing/representation (e.g., the make_examples stage), while the core InceptionV3 backbone has remained unchanged in many releases [10]. More broadly, inference acceleration is often pursued via optimized runtimes (e.g., TensorRT-based inference engines) to improve throughput while aiming to preserve predictive accuracy [11]. Despite this, controlled studies of backbone downsizing and CNN–attention hybrid design in a DeepVariant-style setting remain limited.

We propose a hybrid CNN–local attention architecture that combines convolutional feature extraction with lightweight local context modeling to reduce parameters and FLOPs while maintaining competitive variant-calling accuracy [12]. The primary use case is population-scale genomics and compute-constrained screening workflows, rather than a drop-in replacement for diagnostic-grade pipelines where maximal INDEL precision is required. The design leverages patch-based tokenization and windowed self-attention, which restrict attention to local windows and reduce the quadratic cost of global attention, as established in vision backbones such as Swin Transformer [13]. A related hybrid Transformer–CNN paradigm has also been explored in other genomics tasks (e.g., scDNA-seq denoising) [12]. In our experiments, training time was reduced by ~40–50% while maintaining competitive SNPs/INDELs performance on GIAB short-read benchmarks. In this study, complex regions are defined according to GIAB difficult region categories, including homopolymers, segmental duplications, and low-mappability/low-complexity regions [14,15]. Our benchmarking results are reported as aggregate performance on the full held-out chr20 test set, and we use the GIAB terminology to define the scope of "complex regions" consistently and reproducibly.

This work does not aim to introduce a novel attention mechanism, but rather to empirically characterize the efficiency–accuracy trade-off space of DeepVariant-style models under practical hardware constraints.

In controlled benchmarking under limited computational resources, chromosome-level partitioning is commonly used to reduce direct overlap between splits. However, chromosome-level partitioning may not fully remove positional or chromosome-specific biases, particularly for pileup-based representations.

We address this limitation explicitly and, where feasible, complement chr20 testing with an evaluation on an additional held-out chromosome.

1.1. Contributions

Our main contributions are as follows:

- A controlled architectural efficiency study comparing three DeepVariant-style pileup backbones under identical training and inference conditions: Full InceptionV3 (baseline), InceptionV3 Reduced (CNN-only), and Inc3ViTs (hybrid CNN–local attention).
- An empirical finding that CNN-only downsizing can achieve comparable runtime and competitive SNPs/INDELs performance relative to hybrid designs for short-read GIAB SNPs/INDELs calling, indicating that efficiency gains are largely driven by architectural simplification.
- A practical evaluation of lightweight CNN backbones (EfficientNet-B2), documenting GPU memory limitations on high-resolution multi-channel pileup representations; therefore, EfficientNet-B2 was not included in the final quantitative comparison.

2. Related work

Genomic variant calling has transformed significantly over the last decade, shifting from classical statistical pipelines to advanced machine learning–based approaches [9]. Early tools such as GATK HaplotypeCaller [16], FreeBayes [17], and Strelka2 [18] used probabilistic models, Bayesian inference, and hand-designed heuristics to interpret the evidence produced by sequencing. Although these approaches were state-of-the-art for short-read Illumina data, their accuracy can degrade in challenging contexts such as repetitive and low-mappability regions and loci with ambiguous mapping [8,15]. As largely hand-engineered statistical pipelines, their performance depends on explicit model assumptions and filtering heuristics; reviews comparing statistical and AI-based callers discuss how learned models can better absorb dataset-specific error patterns [9].

A key turning point was the introduction of deep learning approaches, notably DeepVariant, which reframed variant calling as an image classification problem [6]. By transforming read pileups into multi-channel images and using an InceptionV3 CNN [6], DeepVariant replaced human-devised features with learned representations that can learn statistical relationships from images of read pileups around candidate variants, capturing systematic patterns in the aligned-read evidence [6,9]. Across a variety of benchmarks, including GIAB-based

evaluations, DeepVariant demonstrated strong SNP calling performance [6,8]. However, the computational demands of using an InceptionV3 backbone in DeepVariant-style models can be substantial [6,19], and this practical cost is discussed in reviews of AI-based variant calling [9]. Moreover, its limited receptive field may reduce the ability to capture long-range contextual dependencies across reads, particularly in low-mappability or structurally complex regions [20], particularly in low-mappability or structurally complex regions [14].

Hybrid pipelines such as Clair3 were developed to better balance recall and robustness in challenging scenarios [21]. Clair3 integrates several resolutions of evidence by stacking a pileup network and a full-alignment network, followed by an end-to-end refinement module that enables the model to integrate evidence at multiple resolutions [21]. This multi-level design extensions such as Clair3-MP report improved SNP and INDEL F1 in several difficult genomic regions when integrating multi-platform sequencing evidence [22]. However, this type of hybrid architecture can be complex to deploy because it integrates multiple pipelines and increases overall training cost. This multi-stage design may also propagate errors across stages and make accuracy sensitive to upstream filtering or alignment issues [9]. Beyond germline variant calling, the Clair framework has been extended to more specialized contexts. ClairS-TO adapts the Clair framework for tumor-only somatic variant calling (without a matched normal), addressing low-allele-fraction and noise-discrimination challenges [23]. Similarly, Clair3-RNA extends variant calling to RNA-seq data, incorporating transcriptome specific alignment characteristics [24]. These developments highlight the adaptability of deep learning–based architectures to increasingly complex genomic modalities.

Outside classical and hybrid approaches, single-model callers such as Octopus use haplotype-aware probabilistic modeling that integrates local haplotype reconstruction and phasing to improve variant interpretation [25]. Because haplotype-aware probabilistic modeling involves local haplotype reconstruction, runtime can depend on local haplotype complexity [25], and computational trade-offs across callers are discussed in comparative studies and reviews [8,9]. Likewise, Strelka2 emphasizes efficient statistical modeling and filtering for fast germline and somatic calling [18]; reviews and benchmarks discuss context-dependent accuracy –efficiency trade-offs across statistical and deep learning callers [8,9].

However, despite these improvements, important trade-offs remain. CNN-based methods, such as DeepVariant, rely on local convolution operations that may not explicitly capture long-range dependencies between widely separated reads [20]. Multi-stage pipelines, such as Clair3, can improve robustness in challenging scenarios but they typically introduce additional computational and system complexity [9,21]. Conventional statistical callers (e.g., FreeBayes, GATK, Strelka2) may be challenged in difficult genomic contexts due to reliance on hand-designed assumptions and heuristics [8]. Collectively, these observations motivate models that are computationally efficient yet able to learn contextual patterns from short-read pileups under practical computational constraints [8,9,26].

Overall, prior work highlights a recurring trade-off between computational efficiency and robustness in difficult genomic contexts. While attention-based and hybrid designs provide a plausible mechanism to model richer contextual patterns, the extent to which they offer efficiency benefits beyond architectural simplification remains underexplored in DeepVariant-style pileup settings. This motivates a controlled study comparing backbone downsizing and a CNN–local-attention design under identical training and inference conditions, treating the hybrid architecture as one design point rather than as a universally superior replacement.

3. Materials and methods

3.1. Reference data and study design

This study is based on Genome in a Bottle (GIAB) benchmark dataset, which are widely used for assessing the accuracy of small-variant calling [15,27]. To achieve a fair comparison, four reference samples were selected: HG001 (NA12878), HG002, HG003, and HG004, as their variant call sets have undergone in-depth refinement and are frequently used in performance measurement [15,27].

For whole-genome sequencing, we used the publicly released HG001 (NA12878) BGISEQ-500 dataset with paired-end 100 bp reads and approximately 37× coverage [28]. The exome sequence of the Ashkenazi family (HG002, HG003, HG004) were sequenced on the Illumina NovaSeq 6000 platform using IDT xGen Exome Research Panels, producing paired-end 150 bp reads and aligned to the GRCh38 reference with a target depth of about 100× [29].

To reduce leakage across splits, we adopted a chromosome-based partitioning strategy across

training, validation, and test chromosomes. To assess model stability under varying coverage conditions and capture methods, we conducted additional experiments on downsampled HG001 WGS at 21×, 10×, and 6×coverage, as well as on the high-depth exome data for the Ashkenazi trio.

3.2. Chromosome-based data splitting

We followed the official Google DeepVariant training tutorial using chr1 for training, chr21 for validation, and chr20 as a held-out test chromosome. Chromosome 20 (chr20) was selected as the primary held-out test chromosome to balance computational feasibility with evaluation scope, enabling direct comparison with the DeepVariant baseline under identical settings.

We note that, in large-scale production scenarios, training would typically involve multiple chromosomes (e.g., chr1–chr19). The present setup is used to ensure feasibility and reproducibility under constrained computational resources.

In addition to testing on chr20, we report results on an additional held-out chromosome (chr19) that was not used for training or validation, to probe potential chromosome-specific effects in a controlled manner.

3.3. Hardware and execution environment

All experiments were conducted on a single workstation equipped with an NVIDIA GeForce RTX 3070 Ti GPU (8 GB VRAM), an Intel 12th Gen Core i9-12900F CPU (24 cores/48 threads), and 16 GB of system RAM. The software environment consisted of DeepVariant v1.9.0 running inside a Docker container built with CUDA 11.8 and cuDNN 8.9.6.50. Training and inference were performed using TensorFlow 2.13.1 (CUDA-enabled build).

Runtime was measured using end-to-end wall-clock time for the full inference workflow under identical hardware, preprocessing, dataset splits, and batch size settings across all models.

3.4. Baseline configuration and reproducibility

To ensure fair and reproducible benchmarking, all baseline tools were executed using fixed Docker images with explicit version tags. The objective of this study is to compare architectural efficiency under standardized deployment conditions rather than per-tool hyperparameter optimization. No parameter tuning beyond officially recommended pipelines was performed. The pinned containers and modes are reported in the Supplementary File (https://kijoms.uokerbala.edu.iq/cgi/editor.cgi?article=3460&window=additional_files&context=home) (Sheet: Baselines Tools).

3.5. Input representation and preprocessing

Inc3ViTs follows the DeepVariant paradigm by transforming local sequencing evidence into fixed-size multi-channel pileup images summarizing reads around candidate variant sites [6]. Each image encodes base identities, base qualities, mapping qualities, and strand orientation within a predefined genomic window, yielding a tensor $X \in \mathbb{R}^{H\times W\times C}$ that serves as the model input [6], as illustrated in Fig. 1(a).

We used the GIAB-provided aligned BAM files for each sample and reference build and did not re-run alignment. To maintain compatibility with established workflows, candidate variant positions were generated using the candidate generation procedure typical of DeepVariant [6]. Subsequently, we normalized the pileup tensors channel-wise and shuffled samples randomly within each split. It is worth noting that no complex augmentation or label smoothing was necessary; the pileup representation inherently captures the stochastic variability arising from alignment and sequencing noise.

For clarity, the mapping from a raw genomic window $\mathscr{R}$ (reference and aligned reads) to the pileup tensor X can be viewed as a deterministic encoding function

$$X = \Phi(\mathscr{R}),$$

where $\Phi(\cdot)$ combines base encoding, quality scaling, and strand/channel assignment [6].

3.6. Inc3ViTs hybrid architecture

The proposed Inc3ViTs model is a hybrid CNN-Transformer architecture designed to reduce the computational footprint of DeepVariant's InceptionV3 backbone while enhancing its ability to capture local and regional contextual patterns across reads, as depicted in Fig. 1(a). The network operates on the pileup tensor X, which is first processed by the InceptionV3 stem composed of convolution, batch normalization, ReLU activation, and spatial pooling, producing a low-level feature map

$$F_0 = \mathrm{Stem}(X)$$

Subsequently, an Inception Block (Inception Block 1) aggregates multi-scale features in parallel. 1×1, 3×3, 5×5, and pooling branches, yielding

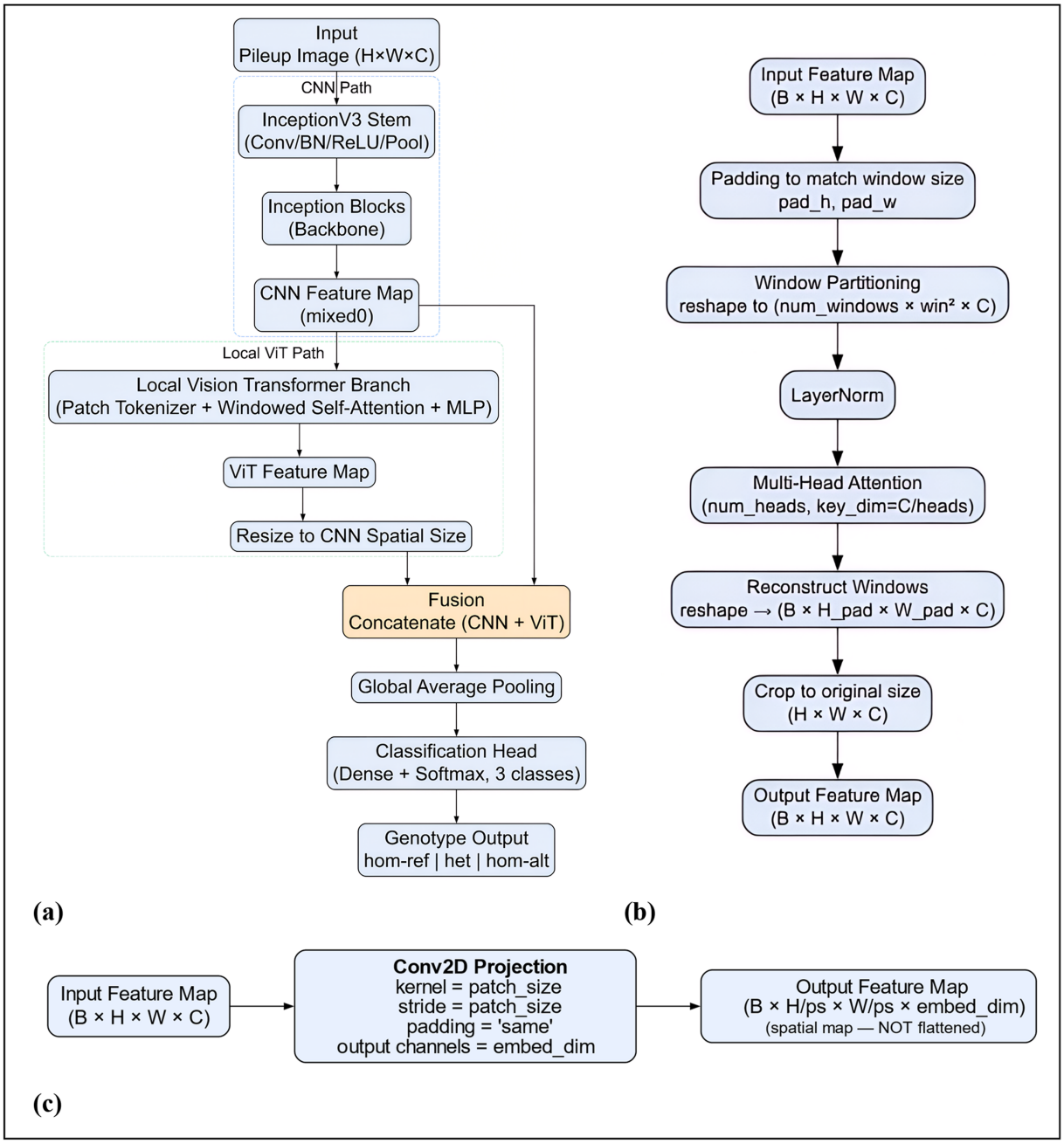

Fig. 1. Inc3ViTs overview: (a) hybrid CNN–ViT architecture, (b) local windowed self-attention, (c) convolutional patch tokenizer.

$F_{\text{mixed0}} = \text{InceptionBlock1}(F_0)$

This tensor constitutes the branching point of the architecture and is forwarded to two processing paths: a convolutional path and a Local-ViT path, as highlighted in Fig. 1(a).

In the CNN path, the model preserves the inductive bias of convolutional processing by directly reusing.

$F_{\text{mixed0}} = F_{\text{CNN}}$

In the Local-ViT path, a Conv2D-based patch tokenizer (Fig. 1(c)) partitions F_{mixed0} into non-overlapping patches with kernel size and stride equal to the patch size p, producing a token grid

$$T = Conv2D\left(F_{mixed0}; kernal = p; stride = p\right), T \in \mathbb{R}^{\frac{H}{p} \times \frac{W}{p} \times D}.$$

The token map is then divided into fixed windows of size. $w \times w$. For each window W_i, a windowed multi-head self-attention (MSA) block

with Layer Normalization is applied, as shown in Fig. 1(b):

$$A_i = \mathrm{MSA}(\mathrm{LN}(W_i)),$$

and the attended windows are reconstructed into a spatial attention feature map.

$$A = \mathrm{Reconstruct}\left(\{A_i\}_{i=1}^{K}\right),$$

where K denotes the number of windows.

A single Dense projection layer refines the attended representation, producing the transformer-enhanced feature map. F_{ViT}. This feature map is then spatially resized to match the resolution of the convolutional feature map. F_{CNN}:

$$F_{\mathrm{ViT}}^{up} = \mathrm{Resize}\left(F_{\mathrm{ViT}}, \mathit{Shape}\,(F_{\mathrm{CNN}})\right),$$

ensuring that both paths can be fused along the channel dimension.

The outputs of the CNN path and the Local-ViT path are fused by channel-wise concatenation to form the unified representation.

$$F_{\mathrm{merged}} = \mathrm{Concat}\left(F_{\mathrm{CNN}}, F_{\mathrm{ViT}}^{up}\right),$$

followed by global average pooling to obtain a compact latent vector

$$z = \mathrm{GAP}\left(F_{\mathrm{merged}}\right).$$

Variant calling is formulated as a three-class classification problem over the genotype states. $\{\text{hom-ref}, \text{het}, \text{hom-alt}\}$.

A fully connected layer with Softmax activation maps the latent representation to genotype probabilities:

$$\widehat{y} = \mathrm{Softmax}(Wz + b),$$

where W and b denote the classifier parameters and $\widehat{y} \in \mathbb{R}^3$ encodes the predicted genotype distribution for each candidate site.

From a computational perspective, the Local-ViT block restricts the self-attention cost to local windows, leading to a complexity of

$$\mathcal{O}\left(K \cdot w^4\right),$$

which is substantially lower than the global attention cost $\mathcal{O}\left(N^2\right)$ with respect to the total number of tokens N. Empirically, this hybridization reduces the FLOPs from approximately. 2.01×10^9 to 0.886×10^9 while preserving variant classification performance.

3.7. InceptionV3 reduced (CNN-only baseline)

To isolate the effect of architectural downsizing independently of attention, we introduce an InceptionV3 Reduced baseline that follows the same DeepVariant-style input and training protocol but removes all tokenization and self-attention components. This baseline was included at the reviewers' request to isolate the effect of architectural downsizing.

Specifically, the model preserves the original InceptionV3 stem up to the first mixed block (mixed0), which constitutes the same branching point used in the proposed Inc3ViTs design. Let the stem output be:

$$F_{\mathrm{mixed0}} = \mathrm{InceptionStem}_{\leq \mathrm{mixed0}}(X)$$

To reduce computational cost, we apply a 1×1 convolution for channel reduction:

$$F_{\mathrm{red}} = \mathrm{Conv}_{1\times1}(F_{\mathrm{mixed0}})$$

followed by a lightweight 3×3 convolutional block (one or more layers depending on the configuration):

$$F_{\mathrm{out}} = \mathrm{Conv}_{3\times3}(F_{\mathrm{red}})$$

Finally, the feature map is pooled using global average pooling and passed to the same genotype classification head (Softmax over $\{\text{hom-ref}, \text{het}, \text{hom-alt}\}$ used in the other architectures.

Importantly, this CNN-only baseline reuses the same input representation, candidate generation, data splits, and evaluation protocol as Inc3ViTs and the full InceptionV3 baseline, enabling a controlled comparison between (i) depth/channel simplification and (ii) hybrid local-attention augmentation.

3.8. Architectural hyperparameters and computational profiling

The final Inc3ViTs hyperparameters (patch size, window size, embedding dimension, attention heads, dropout, and fusion) along with a computed complexity summary are provided in the Supplementary Excel file (https://kijoms.uokerbala.edu.iq/cgi/editor.cgi?article=3460&window=additional_files&context=home) (Sheet: Inc3ViTs_HParams). FLOPs were computed using the TensorFlow 2.13 Profiler API (tf.profiler.experimental) on a single FP32 forward pass (batch size = 1) with the fixed input shape ($100 \times 221 \times 6$), with TensorRT and mixed precision disabled.

3.9. Implementation details and reproducibility

Inc3ViTs was implemented in TensorFlow and integrated into the DeepVariant training pipeline (training binary:/opt/deepvariant/bin/train; base

config: dv_config.py:base). The final training schedule and runtime fairness settings (software versions, hardware, precision settings, and key hyperparameters) are provided in the Supplementary Excel file (https://kijoms.uokerbala.edu.iq/cgi/editor.cgi?article=3460&window=additional_files&context=home) (Sheets: Reproducibility_Environment and Reproducibility details). Peak GPU VRAM and CPU RAM usage were monitored using nvidia-smi and/proc/meminfo, respectively.

3.10. Evaluation protocol

Performance was evaluated on chr20 as the primary held-out test set and on chr19 for cross-chromosome validation, following GIAB/GA4GH small-variant benchmarking best practices [14,30]. For WES, variant calling and benchmarking were restricted to the exome target intervals (BED) and intersected against GIAB high-confidence regions. Metrics included precision, recall, and F1 score reported separately for SNPs and INDELs for aggregate callsets (PASS-only), computed with hap.py (Docker image/version listed in Sheet: Baseline_Tools) [30]. Robustness experiments were additionally conducted on downsampled HG001 WGS (21×/10×/6×) and the Ashkenazi WES trio, under identical preprocessing and evaluation settings.

4. Results

4.1. Training performance

To evaluate efficiency-oriented architectural changes in DeepVariant-style models, we trained the baseline DeepVariant (InceptionV3 backbone) [6,19] and the proposed Inc3ViTs using the same experimental setup. Key training outcomes (wall-clock training time and final accuracy and F1 score) are reported in the main text; the full training metrics and curves are provided in the Supplementary File (https://kijoms.uokerbala.edu.iq/cgi/editor.cgi?article=3460&window=additional_files&context=home) (Sheet: Train).

In our experiments, Inc3ViTs reduced the total training time from 3.60 h to 1.96 h while maintaining high classification accuracy (>0.98) and F1 scores close to those of the baseline model. This reduction in wall-clock time is consistent with the decrease in computational complexity: the standard InceptionV3 backbone requires approximately 2.0×10^9 FLOPs per forward pass, whereas the Inc3ViTs backbone requires approximately 8.9×10^8 FLOPs. Fig. 2 compares the training trajectories of the two models, showing closely aligned accuracy and F1 curves despite the substantial reduction in FLOPs and training time.

4.2. Impact of CNN downsizing versus hybrid attention

To disentangle the sources of the observed efficiency gains, we compare three DeepVariant-style backbones under identical training and inference conditions: (i) the full InceptionV3 baseline, (ii) InceptionV3 Reduced (CNN-only downsizing), and (iii) Inc3ViTs (hybrid CNN–local attention). Table 1 summarizes model complexity (FLOPs), chr20 runtime, and SNPs/INDELs F1-scores (PASS-only).

We observe that the reduced CNN-only baseline achieves runtime and accuracy that are comparable to the hybrid Inc3ViTs model on chr20 (PASS-only). This indicates that most of the efficiency gains are driven by architectural depth/channel reduction in the early backbone rather than the attention mechanism itself under GIAB/GA4GH small-variant benchmarking (PASS-only) [14,30].

We attempted to evaluate EfficientNet-B2 [31] under the same experimental conditions; however, training could not be completed due to higher GPU memory consumption caused by intermediate activation tensors during training. This practical limitation was noted, and EfficientNet-B2 is excluded from quantitative comparison.

4.3. Generalization across chromosomes

To assess whether the observed performance trends are specific to chr20, we additionally evaluated the baseline DeepVariant model and Inc3ViTs on an independent held-out chromosome (chr19), which was not used during training or validation. Chromosome 19 provides a useful cross-chromosome generalization test under the same preprocessing and benchmarking pipeline. The chr19 results showed consistent runtime–accuracy trends relative to those observed on chr20 for these two models, suggesting that the reported behavior is not driven solely by chromosome-specific effects. Table 2 summarizes the cross-chromosome evaluation on an independent held-out chromosome (chr19) using the same training split (train: chr1, val: chr21) and the same benchmarking pipeline. The results reproduce the same runtime–accuracy trend observed on chr20.

4.4. Chromosome 20 WGS evaluation

To provide a comprehensive comparison across all variant callers on the chromosome 20 WGS

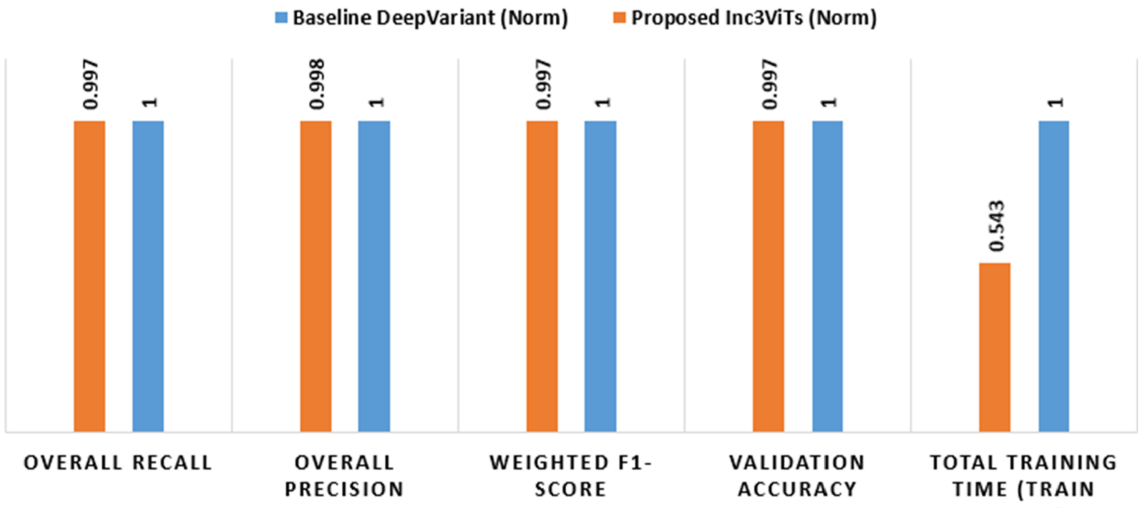

Fig. 2. Normalized comparison of training time and aggregate training metrics for Baseline DeepVariant vs. Inc3ViTs (Baseline = 1.0). Values < 1.0 indicate faster training and/or modest metric reductions.

dataset (BGISEQ_PE100_NA12878), we summarized recall, precision, F1 score, and runtime for INDEL and SNP calls at approximately 37×, 21×, 10×, and 6× coverage. While absolute runtime may vary across different GPU architectures, all comparative evaluations in this study were performed under identical hardware conditions. Therefore, the reported speedup reflects architectural efficiency rather than hardware-specific optimizations.

Peak memory consumption was monitored during both training and inference under identical hardware and batch size settings. Inc3ViTs showed comparable CPU RAM usage and a modest reduction in GPU VRAM consumption (≈1–1.5%) relative to the baseline DeepVariant model. These results suggest that the proposed architectural simplification does not increase memory footprint, while still achieving runtime acceleration.

The resulting CPU RAM and GPU VRAM peaks are summarized in Table 3.

4.4.1. BGISEQ_PE100_NA12878~37× coverage

Table 1 summarizes the ablation study (chr20, PASS-only) across the three backbones in terms of SNP F1 score, INDEL F1 score, runtime, and FLOPs. As shown in Table 1, InceptionV3 Reduced (CNN-only) and Inc3ViTs achieve SNP F1 values close to the baseline DeepVariant (0.996 vs. 0.996), while reducing computational cost substantially (FLOPs: 0.827B–0.887B vs. 2.010B) and slightly improving runtime (≈3 m 47 s–3 m 50 s vs. 4 m 03 s). However, Inc3ViTs shows a modest reduction in INDEL F1 relative to the baseline (0.948 vs. 0.960), whereas the CNN-only reduced model remains closer (0.956). Fig. 3 provides a visual comparison of the corresponding performance trends.

> "Full benchmarking statistics for all callers at 37× coverage are available in Supplementary File 1 (https://kijoms.uokerbala.edu.iq/cgi/editor.cgi?article=3460&window=additional_files&context=home) (Sheet: Inference Chr20 37×)."

Table 1. Ablation: CNN-only downsizing vs. hybrid local attention (chr20, PASS-only).

Model	SNP F1	INDEL F1	Runtime	FLOPs
Baseline DeepVariant	0.996	0.960	4 m 3 s	2,010,121,782
InceptionV3 Reduced (CNN-only)	0.996	0.956	3 m 47 s	827,167,634
Inc3ViTs model	0.995	0.948	3 m 50 s	886,976,640

4.4.2. BGISEQ_PE100_NA12878~ 21× and 10× coverage

At approximately 21× sequencing depth, Inc3ViTs achieves INDELs and SNPs F1 scores that are within 1–2% of the DeepVariant baseline while also reducing inference time from 3.28 min to 3.00 min. Although tools such as GATK, Octopus,

Table 2. Additional held-out chromosome (chr19) results and efficiency metrics.

Model	SNP F1	INDEL F1	Runtime	FLOPs
Baseline DeepVariant	0.994	0.956	4 m 39 s	2,010,361,321
InceptionV3 Reduced (CNN-only)	0.993	0.950	4 m 14 s	827,165,906
Inc3ViTs model	0.987	0.927	3 m 29 s	883,870,272

Table 3. Peak memory consumption comparison.

Phase	Model	Peak CPU RAM (GB)	Peak GPU VRAM (MiB)
Train	Baseline DeepVariant	15.343	5984
Train	Inc3ViTs	15.333	5892
Inference (chr20)	Baseline DeepVariant	12.449	5968
Inference (chr20)	Inc3ViTs	12.385	5918

and Strelka2 obtain slightly higher accuracy, but they do so with substantially greater computational cost. Detailed performance metrics for all callers at this coverage level are available in Supplementary File 1 (https://kijoms.uokerbala.edu.iq/cgi/editor.cgi?article=3460&window=additional_files&context=home) (Sheet: Inference Chr20 21×).

At 10× depth, Inc3ViTs surpasses the baseline DeepVariant model in both INDEL and SNPs F1 scores and maintains faster inference. Its performance at this intermediate coverage remains competitive with Octopus and GATK, whose higher accuracy is accompanied by considerably longer runtimes. The full benchmarking results for the 10× experiments can be found in Supplementary File 1 (https://kijoms.uokerbala.edu.iq/cgi/editor.cgi?article=3460&window=additional_files&context=home) (Sheet: Inference Chr20 10×).

4.4.3. BGISEQ_PE100_NA12878~6× coverage

At the lowest coverage (~6×), the results for all callers are summarized in Table 4. The Inc3ViTs model maintains a favorable balance between accuracy and runtime, improving upon the baseline's F1 scores for both INDELs and SNPs while preserving its status as the fastest neural model, whereas FreeBayes and GATK offer higher INDEL F1 or precision at the expense of runtime.

Fig. 4 shows the INDEL and SNP F1 scores across callers at ~6× coverage, confirming that Inc3ViTs shows improved low-coverage robustness when sequencing depth is limited.

> "Comprehensive low-coverage (~6×) performance metrics are presented in Supplementary File 1 (https://kijoms.uokerbala.edu.iq/cgi/editor.cgi?article=3460&window=additional_files&context=home) (Inference Chr20 6× sheet)."

4.5. Whole exome sequencing (WES) evaluation

We further assessed performance on the HG002, HG003, and HG004 WES datasets with the same seven variant callers. The mean INDEL and SNP F1 scores and mean runtimes across samples are summarized in Table 5, and the F1 scores are visualized in Fig. 5.

As reported in Table 5, the Inc3ViTs model achieves a higher mean SNP F1 score (~0.934) than the DeepVariant baseline (~0.898) while running slightly faster on average, whereas Octopus and GATK achieve the highest overall F1 scores but incur runtimes of approximately 20–45 min per sample. Fig. 5 highlights that Inc3ViTs offers a favorable accuracy–efficiency trade-off for SNP calling in exomes, outperforming the baseline while avoiding the computational overhead associated with GATK.

> "Extended WES benchmarking for samples HG002, HG003, and HG004 is provided in Supplementary File 1 (https://kijoms.uokerbala.edu.iq/cgi/editor.cgi?article=3460&window=additional_files&context=home) (Sheet: WES HG002/HG003/HG004)."

4.6. Statistical significance analysis

To evaluate whether the observed performance differences are statistically meaningful, we conducted a two-proportion z-test on recall and precision metrics using the true positive (TP), false positive (FP), and false negative (FN) counts reported by hap.py. Confidence intervals (95%) were computed for recall and precision, and propagated to F1 score estimation.

The full 95% confidence intervals for key chr20 WGS (~37×, PASS) metrics are reported in Supplementary File 1 (https://kijoms.uokerbala.edu.iq/cgi/editor.cgi?article=3460&window=additional_files&context=home) (Sheet: Stats_CI_37X).

For chr20 WGS (~37×), the INDEL recall difference between DeepVariant (0.9515) and Inc3ViTs (0.9426) corresponds to an absolute gap of 0.89%. Given the total number of true INDEL variants ($N = 10{,}023$), the resulting z-score indicates statistical significance ($p < 0.01$). However, the absolute INDEL F1 reduction remains below 1.3%, while inference runtime improves by ~5% under identical hardware and evaluation settings.

For SNPs calling, performance differences were below 0.2% and fell within confidence interval overlap. These results suggest that the performance trade-off is statistically detectable but practically

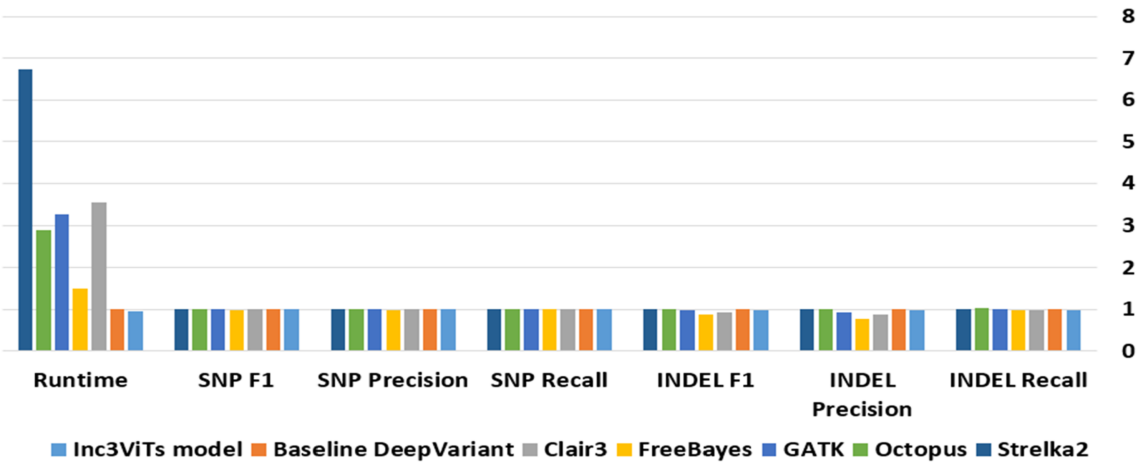

Fig. 3. chr20 WGS (~37×) performance normalized to Baseline DeepVariant (1.0); >1.0 is higher accuracy, <1.0 is faster runtime.

Table 4. Performance comparison of variant callers on chromosome 20 under ultra-low coverage (~6× WGS, PASS only), reporting SNP/INDEL F1-scores and end-to-end inference runtime.

Model	SNP F1	INDEL F1	Runtime
FreeBayes	0.879	0.708	1 m 8 s
Inc3ViTs model	0.759	0.617	1 m 56 s
Baseline DeepVariant	0.716	0.611	2 m 8 s
Octopus	0.866	0.744	3 m 16 s
GATK	0.884	0.742	6 m 41 s
Clair3	0.791	0.675	14 m 55 s
Strelka2	0.829	0.602	19 m 30 s

moderate, particularly when considering the computational efficiency gains.

All significance tests are based on hap.py TP/FP/FN counts on the same GIAB truth set and thus quantify uncertainty with respect to finite sample sizes rather than repeated training runs.

4.7. Error analysis for low-coverage (6×) results

To assess whether the higher recall observed at 6× coverage reflects true signal recovery under sparse evidence rather than an increase in spurious variant calls, we performed a dedicated low-coverage error analysis under the same GIAB benchmarking conditions. This analysis focuses on SNPs Ti/Tv plausibility, false positive characterization by variant type (SNPs vs. INDELs) and error category, and FP/FN decomposition.

4.7.1. Ti/Tv plausibility analysis (SNPs only)

To evaluate the biological plausibility of SNP calls at 6×, we report Ti/Tv ratios for the GIAB truth set and for each caller's SNP query callset as produced by hap.py. Table 6 summarizes the truth Ti/Tv and

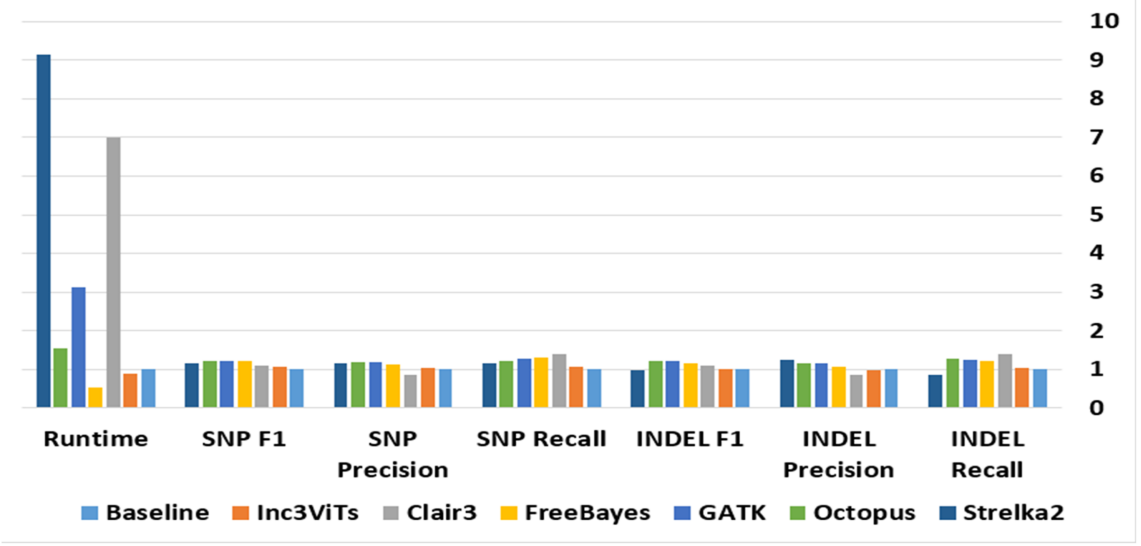

Fig. 4. chr20 WGS (~6×) performance normalized to the baseline (1.0); >1.0 indicates higher accuracy, <1.0 indicates faster runtime.

Table 5. Average INDELs F1, SNPs F1, and runtime across the HG002, HG003, and HG004 WES samples for all evaluated variant callers.

Model	Mean INDEL F1	Mean SNP F1	Mean runtime (min)
Baseline	0.773	0.898	21 m 29 s
Inc3ViTs	0.762	0.934	20 m 30 s
Clair3	0.723	0.943	37 m 54 s
FreeBayes	0.692	0.953	20 m 36 s
GATK	0.819	0.980	45 m 40 s
Octopus	0.906	0.970	20 m 19 s
Strelka2	0.071	0.490	22 m 24 s

the corresponding query Ti/Tv values. Notably, most callers showed query Ti/Tv values close to the truth set, while Clair3 exhibits a higher query Ti/Tv in this setting.

4.7.2. False-positive stratification at 6×

Low-coverage settings can amplify alignment ambiguity and INDEL-related artifacts. Therefore, we stratified false positives at 6× by variant type (SNP vs. INDEL) and by hap.py error category (FP.gt, FP.al) to assess whether the observed low-coverage behavior is accompanied by an increase in false-positive calls (Table 7).

4.7.3. FP/FN error decomposition

To further contextualize performance at 6×, we report a decomposition of errors into FP and FN counts for SNPs and INDELs. This allows distinguishing between "recall gains" driven by reduced FN versus those driven by increased FP, as shown in Table 8.

Overall, these analyses aim to ensure that the low-coverage improvement is supported by biologically plausible SNPs substitution patterns (Ti/Tv) and interpretable error decomposition, rather than reflecting uncontrolled inflation of false-positives calls.

Table 6. Ti/Tv analysis for SNPs calls at 6× coverage (chr20).

Variant Caller	Truth Ti/Tv (SNP)	Query Ti/Tv (SNP)
Baseline DeepVariant	2.284	2.094
Inc3ViTs model	2.284	2.098
FreeBayes	2.284	1.995
Octopus	2.284	2.229
GATK	2.284	2.109
Clair3	2.284	2.848
Strelka2	2.284	2.186

Table 7. False-positive breakdown at 6× coverage (chr20).

Type	Model	QUERY.FP	FP.gt	FP.al
SNP	Baseline DeepVariant	9675	9170	76
SNP	Inc3ViTs	8733	8007	94
INDEL	Baseline DeepVariant	1959	1558	94
INDEL	Inc3ViTs	2097	1591	118

Table 8. FP/FN decomposition at 6× coverage (chr20).

Type	Model	TP (TRUTH.TP)	FP (QUERY.FP)	FN (TRUTH.FN)
SNP	Baseline	42,358	9675	23,879
SNP	Inc3ViTs	45,938	8733	20,299
INDEL	Baseline	5265	1959	4758
INDEL	Inc3ViTs	5400	2097	4623

5. Discussion

This study characterizes efficiency–accuracy trade-offs in DeepVariant-style pileup models.

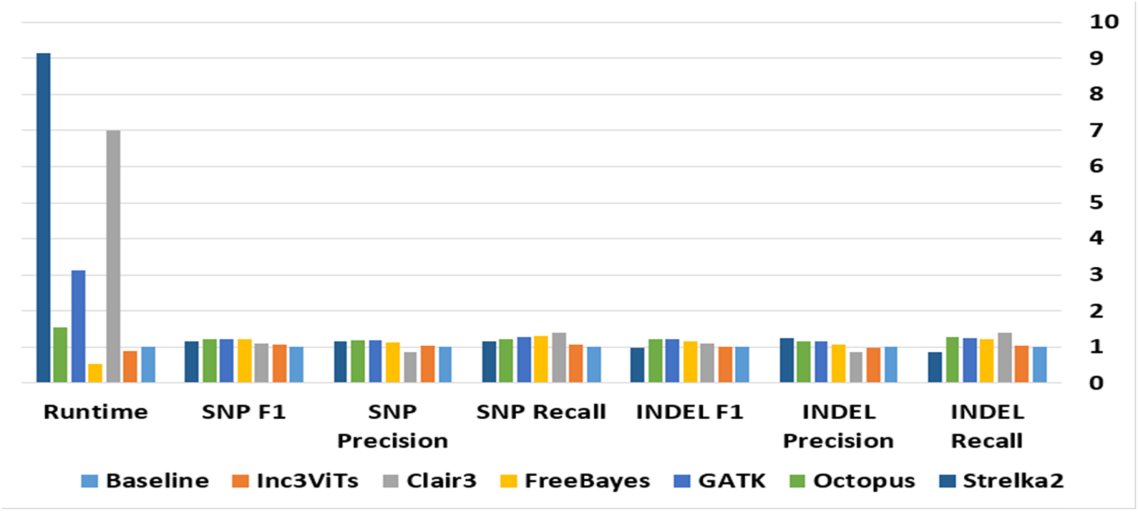

Fig. 5. HG002–HG004 WES mean INDEL F1, mean SNP F1, and mean runtime, normalized to the baseline (1.0); >1.0 indicates higher accuracy, <1.0 indicates faster runtime.

Across WGS (chr20 and an additional held-out chromosome, chr19) and WES (HG002–HG004), Inc3ViTs reduces computational cost and wall-clock time while preserving strong SNP performance, with a modest INDEL precision trade-off at high coverage.

5.1. Limitations of chromosome-level splitting

Chromosome-based splitting reduces direct leakage but does not guarantee full statistical independence. To probe chromosome-specific bias, we additionally evaluated the models on chr19 (excluded from training and validation). The chr19 results reproduced the same runtime–accuracy trend observed on chr20 (faster runtime, strong SNP performance, and an INDEL F1 decrease that is larger on chr19 than on chr20, yet directionally consistent), suggesting that the trade-off is not solely chr20-specific, while noting that INDEL accuracy remains the more sensitive axis under this constrained training regime.

5.2. Training efficiency and model complexity

The most significant performance gain made by Inc3ViTs comes from its much faster training compared with DeepVariant (InceptionV3). This improvement arises from the reduction in computational complexity: replacing most of InceptionV3 with a patch tokenizer and local windowed self-attention results in a lower spatial resolution of learned representations, and it limits attention to windows that can be processed with constant resources [13,32]. Local attention scales approximately as

$$O(N \cdot w^2),$$

in contrast to the

$$O(N^2)$$

complexity of global attention [13,33]. The FLOPs budget is thus reduced from 2.01B to 0.886B, cutting training time from 3.60 h to 1.96 h while preserving strong SNP performance and maintaining high training accuracy, with a modest INDEL trade-off at high coverage.

These enhancements are consistent with results in the computer-vision literature. Some vision literature discusses that CNNs exhibit strong locality/texture inductive biases, whereas transformer-based models may require architectural choices that explicitly strengthen local texture sensitivity [33,34]. InceptionV3, in contrast, is particularly good at capturing fine-grained local spatial patterns and an inductive bias [19,33].

That is helpful for pileup-based variant calling. As a result, the hybrid nature of the Inc3ViTs design retains critical local feature extraction with a lightweight Inception stem and integrates flexible local attention for more accurate context modeling across different tasks.

Crucially, our ablation indicates that a large fraction of the observed efficiency gain is attributable to architectural simplification of the convolutional backbone. In particular, the CNN-only InceptionV3 Reduced baseline achieves chr20 runtime and SNPs and INDELs accuracy that are comparable to Inc3ViTs under the same training and inference conditions, suggesting that depth/channel reduction alone may be sufficient to obtain substantial speedups in this short-read GIAB benchmarking setting. The benefit of attention mechanisms may become more pronounced in other complex scenarios, such as long-read data or structural variant calling, which are outside the scope of this study.

5.3. Speed–accuracy trade-off analysis

At high-coverage WGS (~37×), Inc3ViTs exhibits a modest decrease in INDELs accuracy relative to the DeepVariant baseline (INDELs F1: 0.9482 vs. 0.9609; ≈1.2% percentage points). While statistically significant, the observed INDELs reduction at high coverage should be interpreted as a controlled efficiency–accuracy trade-off rather than an indicator of model instability. Importantly, the effect size remains small relative to the runtime savings, and SNPs performance is preserved. This gap is primarily driven by a small reduction in INDEL precision, with a smaller concurrent decrease in recall, indicating a slight increase in false-positive INDELs calls rather than a loss of sensitivity. From an application standpoint, INDEL precision is often more clinically consequential than SNPs performance, particularly in diagnostic settings where confirmatory testing is costly and minimizing false positives is critical. Accordingly, we do not position Inc3ViTs as a replacement for diagnostic-grade pipelines that require maximal INDEL precision at high coverage. Instead, Inc3ViTs is intended for population-scale genomics and resource-constrained workflows, where runtime and throughput become limiting factors and modest precision differences can be mitigated by downstream filtering, re-scoring, or orthogonal validation when required. This framing clarifies the practical scope of the

proposed design and the intended operating point on the speed–accuracy frontier.

5.4. Effect of sequencing depth on WGS variant calling

5.4.1. High-coverage regimes (37× and 21×)

Among high-depth WGS, all tested callers (including DeepVariant, Inc3ViTs, Octopus, GATK HaplotypeCaller, and Strelka2) have consistently achieved high F1 scores. This is consistent with GIAB benchmarks showing that germline SNP and INDEL calling is very accurate at 25–30× coverage [14,30]. At ~37× and ~21×, Inc3ViTs is near the baseline DeepVariant by a margin of about 1–2%, suggesting that replacing the majority of the convolutional backbone with a lightweight LocalViT branch still maintains DeepVariant-level accuracy while providing an order-of-magnitude reduction in computations.

The small decrease in F1 compared to the complete InceptionV3 backbone is consistent with reports that transformer-based vision models can be less sensitive to fine-grained local texture unless designed to enhance local detail [34], and surveys also note that they often benefit from large-scale training data [33]. However, the difference is small, and it does not justify sacrificing computational efficiency.

5.4.2. Intermediate depth (10×)

The performance of callers diverges more prominently at ~10× coverage. Previous studies have shown that the sensitivity of SNP and INDEL decreases steeply below 15× because there is a lack of sufficient informative reads for reliable reconstruction of haplotype [35]. Despite these difficulties, Inc3ViTs shows an advantage over DeepVariant on both genomes, and SNP F1 is slightly higher at this coverage. These results suggest that combining a lightweight CNN stem with local attention can improve robustness at reduced depth in this setting.

5.4.3. Low coverage (6×)

At 6× coverage, all callers in our experiments experience significant F1 score decreases, reflecting limited evidence at ultra-low depth. Even so, Inc3ViTs remains the fastest of deep-learning–based callers, and maintains better performance than the baseline DeepVariant in both SNP and INDEL F1. When compared to traditional, non-deep learning pipelines, Inc3ViTs shows a more consistent trade-off between accuracy and efficiency than FreeBayes, as well as some of the GATK configurations that experience either longer runtimes (GATK) or less accuracy (FreeBayes).

In our experiments, DeepVariant and Clair3 were among the highest-accuracy callers for short-read variant calling, while Octopus and Strelka2 sometimes achieved comparable or higher F1 values with higher runtimes under the same evaluation settings.

5.5. Behavior of FreeBayes at low coverage

FreeBayes exhibited unusually strong F1 scores and markedly shorter runtimes at 6× coverage; however, this apparent advantage requires cautious interpretation in light of the underlying error profile. A detailed inspection of the confusion metrics in our 6× WGS experiments (Supplementary Inference (https://kijoms.uokerbala.edu.iq/cgi/editor.cgi?article=3460&window=additional_files&context=home) (Sheet: Chr20) 6×)) shows that QUERY.TOTAL substantially exceeds TRUTH.TOTAL for both SNPs and INDELs, consistent with systematic overcalling and a high-recall strategy rather than genuinely improved discrimination. Large UNK (no-call) regions were also observed (Supplementary Inference (https://kijoms.uokerbala.edu.iq/cgi/editor.cgi?article=3460&window=additional_files&context=home) (Sheet: Chr20) 6×), indicating that FreeBayes frequently abstains at sites with weak or ambiguous evidence while aggressively calling at others, thereby inflating recall at the expense of precision. This pattern explains the elevated SNP F1 values at low depth, which are primarily driven by recall rather than balanced accuracy. This observation is concordant with previous comparative benchmarks reporting that FreeBayes generally underperforms DeepVariant, Clair3, Octopus, and Strelka2 for INDEL detection in short-read WGS data [8].

5.6. Generalization to WES data

Whole exome sequencing (WES) introduces distinct challenges, such as non-uniform coverage, GC bias, and capture inefficiencies across targeted exons [36]. Notably, neither DeepVariant nor Inc3ViTs was explicitly trained on WES data. Yet, across HG002, HG003, and HG004, Inc3ViTs improves baseline DeepVariant mean SNPs F1 ($\approx$0.934 vs. $\approx$0.898), while maintaining roughly comparable INDELs performance and achieving faster average runtime.

By contrast, Octopus and GATK achieve the highest INDELs and SNPs F1 scores but require 20–45 min per sample, reflecting the computational

cost of local assembly and haplotype-aware statistical refinement [25,37]. Clair3 and FreeBayes provide competitive SNPs F1 but weaker INDELs performance, while Strelka2 underperforms on WES despite moderate runtimes.

These results indicate that Inc3ViTs offers a particularly advantageous accuracy-efficiency trade-off for SNPs-centric or high-throughput applications, such as population studies, large-scale screening pipelines, or environments with limited computational resources. The hybrid architecture appears to generalize better to the heterogeneous distribution of WES data in our experiments; attributing this behavior to specific architectural factors would require dedicated ablation analyses beyond the scope of this study.

Among evaluated callers, Strelka2 exhibited markedly reduced INDEL and SNP recall across all WES samples. This behavior was consistent across HG002–HG004 and reflects conservative variant emission under default settings and PASS-only filtering. While precision remained high, the limited number of emitted calls resulted in significantly reduced F1 scores. This observation highlights the importance of caller-specific configuration choices when benchmarking WES data.

5.7. Overall perspective

Across a wide range of experiments, the Inc3ViTs architecture demonstrated substantial improvements in computational efficiency and performance. The model reduces training time by approximately 50% and consistently accelerates inference relative to DeepVariant while maintaining competitive accuracy, and in some cases nearly identical performance. It also shows enhanced robustness under reduced sequencing depth, stronger generalization to whole exome sequencing datasets, and remains the fastest neural variant caller across all tested scenarios.

Together, these results highlight that substantial efficiency gains in DeepVariant-style models can be achieved through architectural simplification, with hybrid CNN–local attention representing one feasible design point within this trade-off space. In the evaluated short-read GIAB SNPs/INDELs setting, the CNN-only downsized baseline achieves performance comparable to the hybrid model, indicating that much of the speedup is driven by backbone simplification. Attention mechanisms may provide additional benefits in more complex settings (e.g., long-read or structural variant calling), which are outside the scope of the present study.

Despite these advantages, several limitations remain. The current study focuses primarily on short-read GIAB-derived WGS and WES data, with training restricted to selected chromosomes. Future research should explore the model's applicability across more diverse populations, sequencing technologies, and variant types, particularly structural variants and long-read platforms. Further investigations into dynamic attention window sizing, genome-wide training strategies, and the incorporation of haplotype-aware features may enhance performance. Validation within clinical pipelines and large-scale production environments will also be important to establish the model's robustness in real-world applications.

6. Conclusion

This study provides a practical and empirical evaluation of efficiency-oriented architectural choices for DeepVariant-style models, using Inc3ViTs as a representative hybrid CNN–local attention design alongside CNN-only downsizing baselines. Across the evaluated architectures, we observe substantial reductions in FLOPs and wall-clock training and inference time relative to the full InceptionV3 baseline. Most efficiency gains are attributable to architectural depth/channel simplification in the early backbone, with additional effects from the hybrid local-attention branch that are variable across settings (and modest in some GIAB short-read configurations). Empirical results demonstrate a reduction of approximately 40–50% in training time and consistently faster inference while maintaining competitive SNPs and INDELs performance in the evaluated GIAB short-read setting, highlighting that architecture-level simplification can meaningfully improve efficiency beyond system-level acceleration under controlled and comparable experimental conditions.

We also observe no memory overhead, peak CPU RAM is comparable, and peak GPU VRAM is slightly reduced relative to the baseline.

Across the evaluated GIAB short-read WGS and WES benchmarks, Inc3ViTs achieves accuracy that is generally comparable to the DeepVariant baseline, with performance differences that depend on variant type and coverage regime. Inc3ViTs achieves competitive accuracy for short-read SNP and INDEL variant calling under GIAB benchmarking conditions, while delivering consistently faster inference relative to the DeepVariant baseline. On the primary chr20 high-coverage tests (37× and 21×), the model retains SNP and INDEL F1 scores within ~1–2% of the baseline, despite the simplified

backbone; on the cross-chromosome chr19 test we observe a larger INDEL gap. At intermediate and low coverage depths (10×–6×), Inc3ViTs shows improved robustness relative to the baseline in several settings, while maintaining competitive performance compared with alternative callers that may rely on more computationally expensive heuristics, such as haplotype assembly or full-alignment refinement. On WES datasets (not used for training), Inc3ViTs shows competitive SNPs performance with reduced runtime relative to the baseline in our experiments.

Comparison against a variety of traditional and deep learning-based callers, including Clair3, GATK HaplotypeCaller, FreeBayes, Octopus, and Strelka2, further demonstrates the competitive accuracy-efficiency trade-off reached by Inc3ViTs. Although some tools produce slightly better F1 scores in certain situations, they do so at the cost of substantially longer runtimes. On the other hand, Inc3ViTs consistently reduces inference time relative to the DeepVariant baseline across the tested scenarios, supporting its suitability for population-scale studies and compute-constrained screening workflows, where throughput is a primary constraint and downstream filtering or validation can be applied when needed.

Inc3ViTs is not intended to replace diagnostic-grade pipelines that prioritize maximal INDEL precision at high coverage; rather, it provides a speed-efficient operating point for large-cohort and resource-limited settings.

The evaluation presented in this study is limited to Genome in a Bottle (GIAB) benchmark datasets using short-read sequencing data and small variant calling (SNPs and short INDELs). Structural variants, large insertions/deletions, copy-number variations, and long-read sequencing technologies (e.g., PacBio or Oxford Nanopore) were not investigated. Therefore, conclusions should not be generalized beyond small-variant short-read variant calling scenarios.

Performance may be further improved by extending training to genome-wide datasets, incorporating haplotype-aware representations, and investigating adaptive or dynamic attention mechanisms. Validation in clinical or production-scale workflows will also be important to assess robustness under operational constraints.

In summary, our results indicate that substantial efficiency gains in DeepVariant-style models can be achieved through architecture-level simplification, with hybrid CNN–local attention representing one feasible design point within this efficiency–accuracy trade-off space. In the evaluated GIAB short-read SNPs/INDELs setting, Inc3ViTs provides a practical balance between runtime and accuracy, supporting its potential use in large-scale and resource-constrained workflows.

Ethical approval

This study did not involve any new experiments on human participants or animals. All genomic data analyzed in this research were obtained from a publicly available, high-confidence Genome in a Bottle (GIAB) dataset. Therefore, ethical approval was not required, and this section is deemed not applicable.

Funding

No external funding was received for this study.

Code availability

The source code is publicly available in the repository https://github.com/mustafamax/deepvariant-Inc3ViTs-Model-.

Conflicts of interest

The authors report there are no competing interests to declare.

Acknowledgements

The authors acknowledge the College of Information Technology at the University of Babylon for their academic support. They also wish to thank the other colleagues for their comments and help with the languages of the manuscript.

Supplementary Materials.

The supplementary material is provided as a single Excel file compiling detailed training and inference results across multiple datasets and sequencing depths. It reports a side-by-side training comparison between the baseline DeepVariant (InceptionV3) and the proposed Inc3ViTs model, and provides comprehensive variant-calling evaluations on whole-genome sequencing (WGS) at approximately ~37×, ~21×, ~10×, and ~6× coverage. Additional worksheets summarize whole exome sequencing (WES) performance across HG002, HG003, and HG004, enabling assessment of robustness and generalization under distinct sequencing conditions.

The supplementary file also includes an independent held-out chromosome evaluation on chr19.

Chromosome 19 was strictly excluded from both training and validation and was used only for final testing to assess cross-chromosome generalization. All experimental conditions for chr19 (hyperparameters, hardware/software environment, input representation, and the hap.py/vcfeval-based evaluation pipeline) were kept identical to the main chr20 experiments to ensure a fair and directly comparable assessment. For clarity, the Excel file is organized into the following sheets: Train, Chr20_37x, Chr20_21x, Chr20_10x, Chr20_6x, Stats_CI_37x, WES_HG002, WES_HG003, WES_HG004, Chr19, Reproducibility_details, Baseline_Tools, and Inc3ViTs_HParams/Reproducibility_Environment.

References

[1] W.R. McCombie, J.D. McPherson, E.R. Mardis, Next-generation sequencing technologies, cold spring harb, Perspect. Med. 9 (11) (2019) 1–8, https://doi.org/10.1101/cshperspect.a036798.

[2] H. Satam, K. Joshi, U. Mangrolia, S. Waghoo, G. Zaidi, S. Rawool, R.P. Thakare, S. Banday, A.K. Mishra, G. Das, S.K. Malonia, Next-generation sequencing technology: current trends and advancements, Biology (Basel) 12 (7) (2023) 1–25, https://doi.org/10.3390/biology12070997.

[3] K.R.M. Shaw, A. Maitra, The status and impact of clinical tumor genome sequencing, Annu. Rev. Genomics Hum. Genet. 20 (2019) 413–432, https://doi.org/10.1146/annurev-genom-083118-015034.

[4] Ş. Ari, M. Arikan, Next-generation sequencing: advantages, disadvantages, and future, in: K.R. Hakeem, H. Tombuloğlu, G. Tombuloğlu, eds., Plant Omics: Trends and Applications, Springer International Publishing, Cham. 2016, pp. 109–135, https://doi.org/10.1007/978-3-319-31703-8_5.

[5] F.O. Bagger, L. Borgwardt, A.S. Jespersen, A.R. Hansen, B. Bertelsen, M. Kodama, F.C. Nielsen, Whole genome sequencing in clinical practice, BMC Med. Genomics 17 (1) (2024) 1–16, https://doi.org/10.1186/S12920-024-01795-W.

[6] R. Poplin, P.C. Chang, D. Alexander, S. Schwartz, T. Colthurst, A. Ku, D. Newburger, J. Dijamco, N. Nguyen, P. T. Afshar, S.S. Gross, L. Dorfman, C.Y. McLean, M.A. DePristo, A universal SNP and small-indel variant caller using deep neural networks, Nat Biotechnol 36 (10) (2018) 983–987, https://doi.org/10.1038/nbt.4235.

[7] E. Ruark, E. Holt, A. Renwick, M. Münz, M. Wakeling, S. Ellard, S. Mahamdallie, S. Yost, N. Rahman, ICR142 benchmarker: evaluating, optimising and benchmarking variant calling performance using the ICR142 NGS validation series, Wellcome Open Res 3 (108) (2018) 1–18, https://doi.org/10.12688/wellcomeopenres.14754.2.

[8] Y.A. Barbitoff, R. Abasov, V.E. Tvorogova, A.S. Glotov, A. V. Predeus, Systematic benchmark of state-of-the-art variant calling pipelines identifies major factors affecting accuracy of coding sequence variant discovery, BMC Genom 23 (1) (2022) 1–17, https://doi.org/10.1186/S12864-022-08365-3.

[9] O. Abdelwahab, D. Torkamaneh, Artificial intelligence in variant calling: a review, Front. Bioinform. 5 (1574359) (2025) 1–10, https://doi.org/10.3389/FBINF.2025.1574359.

[10] A. Gurianova, A. Pestruilova, A. Beliaeva, A. Kasianov, L. Mikhailova, E. Guguchkin, E. Karpulevich, Rethinking DeepVariant: efficient neural architectures for intelligent variant calling, Int J Mol Sci 27 (1) (2026) 1–15, https://doi.org/10.3390/ijms27010513.

[11] I.J. Ratul, Y. Zhou, K. Yang, Accelerating deep learning inference: a comparative analysis of modern acceleration frameworks, Electronics 14 (15) (2025) 1–20, https://doi.org/10.3390/electronics14152977.

[12] Z. Yu, F. Liu, Y. Li, scTCA: a hybrid transformer-CNN architecture for imputation and denoising of scDNA-seq data, Brief. Bioinform. 25 (6) (2024) 1–12, https://doi.org/10.1093/bib/bbae577.

[13] Z. Liu, Y. Lin, Y. Cao, H. Hu, Y. Wei, Z. Zhang, S. Lin, B. Guo, Swin transformer: hierarchical vision transformer using shifted windows, in: Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV), IEEE. 2021, pp. 9992–10002, https://doi.org/10.1109/ICCV48922.2021.00986.

[14] J.M. Zook, J. McDaniel, N.D. Olson, J. Wagner, H. Parikh, H. Heaton, S.A. Irvine, L. Trigg, R. Truty, C.Y. McLean, F.M. De La Vega, C. Xiao, S. Sherry, M. Salit, An open resource for accurately benchmarking small variant and reference calls, Nat Biotechnol 37 (5) (2019) 561–566, https://doi.org/10.1038/s41587-019-0074-6.

[15] J. Wagner, N.D. Olson, L. Harris, Z. Khan, J. Farek, M. Mahmoud, A. Stankovic, V. Kovacevic, B. Yoo, N. Miller, J. A. Rosenfeld, B. Ni, S. Zarate, M. Kirsche, S. Aganezov, M. C. Schatz, G. Narzisi, M. Byrska-Bishop, W. Clarke, U.S. Evani, C. Markello, K. Shafin, X. Zhou, A. Sidow, V. Bansal, P. Ebert, T. Marschall, P. Lansdorp, V. Hanlon, C.A. Mattsson, A.M. Barrio, I.T. Fiddes, C. Xiao, A. Fungtammasan, C.S. Chin, A.M. Wenger, W.J. Rowell, F.J. Sedlazeck, A. Carroll, M. Salit, J.M. Zook, Benchmarking challenging small variants with linked and long reads, Cell Genom 2 (5) (2022) 1–8, https://doi.org/10.1016/J.XGEN.2022.100128.

[16] A. McKenna, M. Hanna, E. Banks, A. Sivachenko, K. Cibulskis, A. Kernytsky, K. Garimella, D. Altshuler, S. Gabriel, M. Daly, M.A. DePristo, The genome analysis toolkit: a mapreduce framework for analyzing next-generation DNA sequencing data, Genome Res 20 (9) (2010) 1297–1303, https://doi.org/10.1101/GR.107524.110.

[17] H. Li, Toward better understanding of artifacts in variant calling from high-coverage samples, Bioinformatics 30 (20) (2014) 2843–2851, https://doi.org/10.1093/bioinformatics/btu356.

[18] S. Kim, K. Scheffler, A.L. Halpern, M.A. Bekritsky, E. Noh, M. Källberg, X. Chen, Y. Kim, D. Beyter, P. Krusche, C.T. Saunders, Strelka2: fast and accurate calling of germline and somatic variants, Nat Methods 15 (8) (2018) 591–594, https://doi.org/10.1038/S41592-018-0051-X.

[19] C. Szegedy, V. Vanhoucke, S. Ioffe, J. Shlens, Z. Wojna, Rethinking the inception architecture for computer vision, in: Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR), IEEE. 2016, pp. 2818–2826, https://doi.org/10.1109/CVPR.2016.308.

[20] L. Alzubaidi, J. Zhang, A.J. Humaidi, A. Al-Dujaili, Y. Duan, O. Al-Shamma, J. Santamaría, M.A. Fadhel, M. Al-Amidie, L. Farhan, Review of deep learning: concepts, CNN architectures, challenges, applications, future directions, J Big Data 8 (1) (2021) 1–74, https://doi.org/10.1186/S40537-021-00444-8.

[21] Z. Zheng, S. Li, J. Su, A.W.S. Leung, T.W. Lam, R. Luo, Symphonizing pileup and full-alignment for deep learning-based long-read variant calling, Nat Comput Sci 2 (12) (2022) 797–803, https://doi.org/10.1038/s43588-022-00387-x.

[22] H. Yu, Z. Zheng, J. Su, T.W. Lam, R. Luo, Boosting variant-calling performance with multi-platform sequencing data using Clair3-MP, BMC Bioinf 24 (1) (2023) 1–21, https://doi.org/10.1186/S12859-023-05434-6.

[23] L. Chen, Z. Zheng, J. Su, X. Yu, A.O.K. Wong, J. Zhang, Y.L. Lee, R. Luo, ClairS-TO: a deep-learning method for long-read tumor-only somatic small variant calling, Nat Commun 16 (1) (2025) 1–15, https://doi.org/10.1038/s41467-025-64547-z.

[24] Z. Zheng, X. Yu, L. Chen, Y.L. Lee, C. Xin, A.O.K. Wong, M. Jain, R.K. Kesharwani, F.J. Sedlazeck, R. Luo, Clair3-RNA: a deep learning-based small variant caller for long-read RNA

sequencing data, Nat Commun 16 (1) (2025) 1–19, https://doi.org/10.1038/s41467-025-67237-y.
[25] D.P. Cooke, D.C. Wedge, G. Lunter, A unified haplotype-based method for accurate and comprehensive variant calling, Nat Biotechnol 39 (7) (2021) 885–892, https://doi.org/10.1038/S41587-021-00861-3.
[26] F. Hanssen, M.U. Garcia, L. Folkersen, A.S. Pedersen, F. Lescai, S. Jodoin, E. Miller, M. Seybold, O. Wacker, N. Smith, G. Gabernet, S. Nahnsen, Scalable and efficient DNA sequencing analysis on different compute infrastructures aiding variant discovery, NAR Genom. Bioinform 6 (2) (2024) 1–14, https://doi.org/10.1093/NARGAB/LQAE031.
[27] J.M. Zook, D. Catoe, J. McDaniel, L. Vang, N. Spies, A. Sidow, Z. Weng, Y. Liu, C.E. Mason, N. Alexander, E. Henaff, A.B.R. McIntyre, D. Chandramohan, F. Chen, E. Jaeger, A. Moshrefi, K. Pham, W. Stedman, T. Liang, M. Saghbini, Z. Dzakula, A. Hastie, H. Cao, G. Deikus, E. Schadt, R. Sebra, A. Bashir, R.M. Truty, C.C. Chang, N. Gulbahce, K. Zhao, S. Ghosh, F. Hyland, Y. Fu, M. Chaisson, C. Xiao, J. Trow, S.T. Sherry, A.W. Zaranek, M. Ball, J. Bobe, P. Estep, G.M. Church, P. Marks, S. Kyriazopoulou-Panagiotopoulou, G.X.Y. Zheng, M. Schnall-Levin, H.S. Ordonez, P.A. Mudivarti, K. Giorda, Y. Sheng, K.B. Rypdal, M. Salit, Extensive sequencing of seven human genomes to characterize benchmark reference materials, Sci Data 3 (160025) (2016) 1–26, https://doi.org/10.1038/sdata.2016.25.
[28] J. Huang, X. Liang, Y. Xuan, C. Geng, Y. Li, H. Lu, S. Qu, X. Mei, H. Chen, T. Yu, N. Sun, J. Rao, J. Wang, W. Zhang, Y. Chen, S. Liao, H. Jiang, X. Liu, Z. Yang, F. Mu, S. Gao, A reference human genome dataset of the BGISEQ-500 sequencer, GigaScience 6 (5) (2017) 1–9, https://doi.org/10.1093/GIGASCIENCE/GIX024.
[29] C. Bhérer, R. Eveleigh, K. Trajanoska, J. St-Cyr, A. Paccard, P. Nadukkalam Ravindran, E. Caron, N. Bader Asbah, P. McClelland, C. Wei, I. Baumgartner, M. Schindewolf, Y. Döring, D. Perley, F. Lefebvre, P. Lepage, M. Bourgey, G. Bourque, J. Ragoussis, V. Mooser, D. Taliun, A cost-effective sequencing method for genetic studies combining high-depth whole exome and low-depth whole genome, NPJ Genom. Med. 9 (1) (2024) 1–12, https://doi.org/10.1038/s41525-024-00390-3.
[30] P. Krusche, L. Trigg, P.C. Boutros, C.E. Mason, F.M. De La Vega, B.L. Moore, M. Gonzalez-Porta, M.A. Eberle, Z. Tezak, S. Lababidi, R. Truty, G. Asimenos, B. Funke, M. Fleharty, B. A. Chapman, M. Salit, J.M. Zook, Best practices for benchmarking germline small-variant calls in human genomes, Nat Biotechnol 37 (5) (2019) 555–560, https://doi.org/10.1038/S41587-019-0054-X.
[31] M. Tan, Q.V. Le, EfficientNet: rethinking model scaling for convolutional neural networks, in: K. Chaudhuri, R. Salakhutdinov, eds., Proc. 36th Int. Conf. Mach. Learn. (ICML), Proc. Mach. Learn. Res. 97, 2019, pp. 6105–6114, https://doi.org/10.48550/arXiv.1905.11946.
[32] Y.H. Wu, S.C. Zhang, Y. Liu, L. Zhang, X. Zhan, D. Zhou, J. Feng, M.M. Cheng, L. Zhen, Low-resolution self-attention for semantic segmentation, IEEE Trans Pattern Anal Mach Intell 47 (9) (2025) 8180–8192, https://doi.org/10.1109/TPAMI.2025.3577035.
[33] A. Khan, Z. Rauf, A. Sohail, A.R. Khan, H. Asif, A. Asif, U. Farooq, A survey of the vision transformers and their CNN-transformer based variants, Artif Intell Rev 56 (Suppl 3) (2023) 2917–2970, https://doi.org/10.1007/s10462-023-10595-0.
[34] R. Azad, A. Kazerouni, B. Azad, E. Khodapanah Aghdam, Y. Velichko, U. Bagci, D. Merhof, Laplacian-former: overcoming the limitations of vision transformers in local texture detection, in: H. Greenspan, A. Madabhushi, P. Mousavi, S. Salcudean, J. Duncan, T. Syeda-Mahmood, R. Taylor, eds., Medical Image Computing and Computer-Assisted Intervention – MICCAI 2023, Lect. Notes Comput. Sci. 14222, Springer Nature Switzerland, Cham. 2023, pp. 736–746, https://doi.org/10.1007/978-3-031-43898-1_70.
[35] N. Rieber, M. Zapatka, B. Lasitschka, D. Jones, P. Northcott, B. Hutter, N. Jäger, M. Kool, M. Taylor, P. Lichter, S. Pfister, S. Wolf, B. Brors, R. Eils, Coverage bias and sensitivity of variant calling for four whole-genome sequencing technologies, PLoS One 8 (6) (2013) 1–11, https://doi.org/10.1371/JOURNAL.PONE.0066621.
[36] Q. Wang, C.S. Shashikant, M. Jensen, N.S. Altman, S. Girirajan, Novel metrics to measure coverage in whole exome sequencing datasets reveal local and global non-uniformity, Sci Rep 7 (1) (2017) 1–11, https://doi.org/10.1038/s41598-017-01005-x.
[37] M.A. DePristo, E. Banks, R. Poplin, K.V. Garimella, J.R. Maguire, C. Hartl, A.A. Philippakis, G. Del Angel, M.A. Rivas, M. Hanna, A. McKenna, T.J. Fennell, A.M. Kernytsky, A.Y. Sivachenko, K. Cibulskis, S.B. Gabriel, D. Altshuler, M. J. Daly, A framework for variation discovery and genotyping using next-generation DNA sequencing data, Nat Genet 43 (5) (2011) 491–498, https://doi.org/10.1038/NG.806.