Computing (2025) 107:71
https://doi.org/10.1007/s00607-025-01427-w

REGULAR PAPER

Approximating vision transformers for edge: variational inference and mixed-precision for multi-modal data

Dewant Katare[1] · **Sam Leroux**[2] · **Marijn Janssen**[1] · **Aaron Yi Ding**[1]

Received: 25 September 2024 / Accepted: 22 January 2025

Abstract

Vision transformer (ViTs) models have shown higher accuracy, robustness and large volume data processing ability, creating new baselines and references for perception tasks. However, these advantages require large memory and high-performance processors and computing units, which makes model adaptability and deployment challenging within resource-constrained environments such as memory-restricted and battery-powered edge devices. This paper addresses the model deployment challenges by proposing a model approximation approach **VI-ViT**, for edge deployment using variational inference with mixed precision for processing multi-modalities, such as point clouds and images. Our experimental evaluation on the nuScenes and Waymo datasets show up to 37% and 31% reduction in model parameters and Flops while maintaining a mean average precision of 70.5 compared to 74.8 of the baseline model. This work presents a practical deployment approach for approximating and optimizing Vision Transformers for edge AI applications by balancing model metrics such as parameters, flops, latency, energy consumption, and accuracy, which can easily be adapted to other transformer models and datasets.

Keywords Mixed precision · Model approximation · Variational parameters · Vision transformers · Edge AI · Quantization · Multimodality

✉ Dewant Katare
d.katare@tudelft.nl

Sam Leroux
sam.leroux@ugent.be

Marijn Janssen
m.f.w.h.a.janssen@tudelft.nl

Aaron Yi Ding
aaron.ding@tudelft.nl

[1] Department of Engineering Systems and Services, Delft University of Technology, Jaffalaan 5, 2628 BX Delft, The Netherlands

[2] IDLab, Department of Information Technology at Ghent University - imec, Technologiepark-Zwijnaarde 126, 9052 Gent, Belgium

Published online: 13 February 2025

1 Introduction

The development of large vision models has improved baseline model performance and high-volume data handling capabilities [1]. Vision Transformers (ViTs) have shown benchmark performance for tasks such as classification and complex scene understanding due to their effective architecture and multi-head attention mechanism, which supports a range of complex perception tasks, including image segmentation and localization [2–6] (a representative architecture and processing pipeline of a multi-modality vision transformer is shown in Fig. 1). The fusion of point cloud and camera data helps to achieve accurate localization and precise classification in use cases such as autonomous vehicles. Combining these two modalities improves the model's understanding of the environment, offering a more robust and accurate perception in diverse conditions [7]. The scaled deployment of these models is challenging because of high computational, memory, and energy demands [1, 5, 8, 9]. The deployment complexity further increases for use cases such as autonomous driving and intelligent traffic monitoring systems [5, 10], which require partial data processing and computation on the edge (e.g., edge assisted High-definition map update) rather than entirely on the cloud [11]. From the communication perspective, the high data throughput of these models can also result in additional power requirements on the edge device [12]. Thus, a balanced deployment using only edge or cloud-edge collaboration requires economic and environmental considerations. Economically, the cost of operating high-performance computing units in a distributed environment becomes unsustainable [12]. Environmentally, the high energy consumption and resulting carbon footprint require an adaptation of sustainable and green AI practices [11, 13].

The transformer model's dense parameterization requires considerable onboard memory for operation, thus requiring a compressed and advanced optimization process for efficient model deployment on the edge, which has limited onboard memory

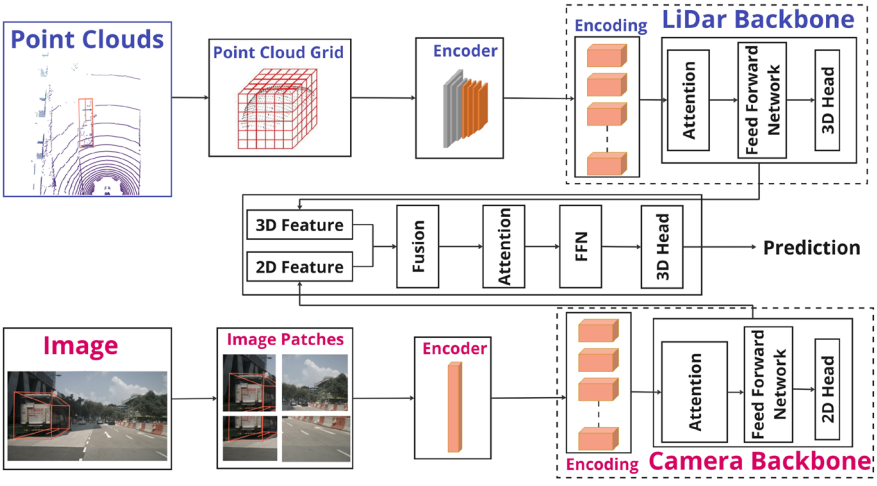

Fig. 1 Pipeline for multi-modality Transformer [6, 8]

and compute/processing power, as these devices are customized for efficiency and compactness [14, 15]. Popular compression and optimization techniques (a categorization of these techniques is shown in Fig. 2) such as normalization, model pruning, sparsification, and dimensionality reduction have resulted in compressed transformer models by reducing model parameters and weights, thereby reducing memory and compute demands while training and deploying models [5, 14–19].

Although the above methods produce optimized models with compressed architectures, they do not directly address issues such as high energy or power consumption from the device. The self-attention mechanisms of transformers still require considerable processing power, and latency issues affect real-time data processing capabilities [16]. Additionally, compressing models leads to accuracy losses, affecting performance in essential applications like autonomous driving. To address the above-mentioned challenges, this paper focuses on balanced trade-off mechanisms using variational inference and mixed-precision quantization for developing and deploying ViTs in resource-constrained environments (Our targeted optimization is also highlighted in Fig. 2). To address real-world complexity, the focus is given to multi-modal data processing applications such as object detection and classification using point clouds and images. Our research exploration and contributions in VI-ViT are as follows:

1. Adaptive variational inference strategy combined with mixed-precision quantization to optimize vision transformers for efficient multi-modal data processing.
2. An approximate inference training algorithm for point cloud and image data to strike a balance between energy efficiency, accuracy and computing demands.
3. Optimized joint-loss function to balance quantization loss and model performance loss, and method evaluation using the multi-distribution mechanism.

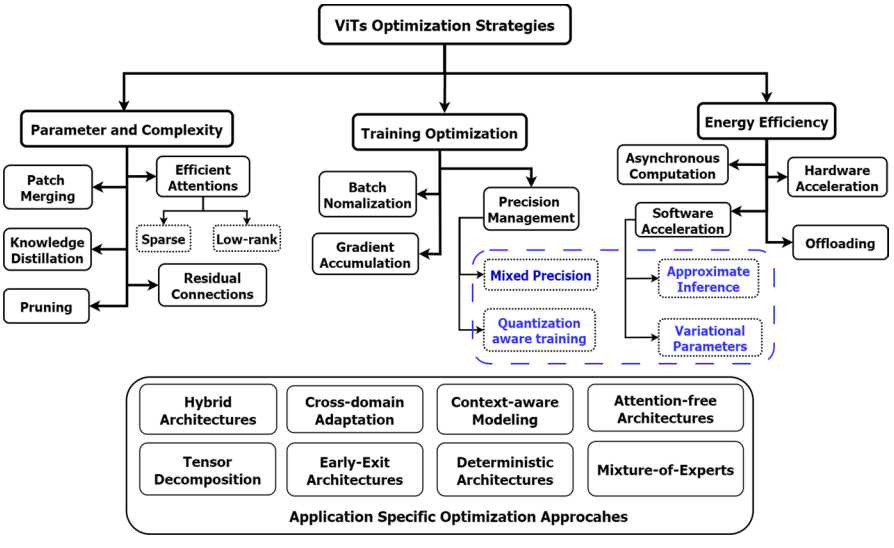

Fig. 2 Optimization Techniques for Vision Transformers

While reducing model parameters and compressing the model, our proposed approach[1] also improves and balances computational efficiency, energy consumption, and model accuracy. The proposed approach includes an algorithm which combines mixed-precision and variational inference for multiple complex components for the transformer model. The method and strategy prove effective in developing an optimized lightweight model for edge AI applications. For comparison with baselines, we use models such as RangeViT [20], TransFusion [8] and Mobile-ViT [10], on the popular nuScenes and Waymo driving datasets by maintaining comparable training and testing splits. We use average precision (AP), mean average precision (mAP) and intersection-over-union (IoU) for model performance metrics. Flops, latency and energy are used as device performance metrics for comparison. We use speedup and the model's respective energy consumption device for model and device correlation.

2 Background and related work

Vision Transformers (ViTs) architectures are efficient in capturing long-range dependence (LRD) and are capable of handling high-volume data as compared to the CNN and DNN models [21]. The architecture is inspired by self-attention mechanisms initially proposed for language processing architectures (transformers) to address the challenges and problems in the encoder-decoder architectures, which were previous baseline architecture [22]. The fundamentals of transformer architecture include patch embedding and feature extraction using self-attention blocks. The architecture divides the input image into patches to form embedded sequences for a classification task [21]. The encoder block then processes these embedded sequences, using multiple self-attention layers extracting low-level and high-level features from the data. A multi-layer perceptron (MLP) generates output probabilities at the final stage. This combination allows ViTs to perform well on tasks like object classification, detection, and segmentation [1, 2, 8, 14, 16, 21–24]. To improve the performance, hybrid approaches combining ViTs with convolutional neural networks (CNNs) have been proposed [1, 16, 22]. These hybrid methods utilize the CNNs and DNNs modules to improve local information processing while benefiting from the ViTs' ability to model long-range dependencies. Such methods have resulted in hybrid architectures where transformer blocks are replaced with fully connected CNN. Similarly, CNN's initial layers are modified with attention mechanisms to capture global features and enhance overall model performance.

Extending the application of ViTs from image-based tasks to processing point clouds introduces memory and computing complexities because of additional data processing. Existing transformers in this category include FusionViT [6], and Point-BERT [25], which depend on farthest point sampling and the K-nearest-neighbour algorithm to generate input tokens. Other popular architecture using point cloud includes PointTransformer [26] and BridgedTransformer [27],

[1] Project and code can be accessed at: VI-VIT.

which introduces a self-supervised pretraining approach, inspired by the U-Net architectures [16, 28]. These models and techniques have introduced new families or series of vision transformers capable of processing and fusing multi-modalities for applications that require segmentation, localization or detection using multiple types of data.

Datasets like nuScenes and Waymo are standards in the autonomous driving domain [29, 30], providing diverse scenarios for rigorous testing and benchmarking of models. Comparative studies using platforms like OpenPCDet offer insights into the performance of various models and techniques in perception tasks [31]. Such studies with balanced metrics approximation are essential for understanding solutions and identifying improvement areas. These datasets and SOTA models (CNN, DNNs) have been comprehensively covered in previous works [8, 17, 31–33]. As this paper focuses on ViTs, the following subsections discuss compression and optimization techniques for vision transformers, followed by variational inference and mixed-precision quantization, which is also categorized in Fig. 2. The figure shows strategies for Vision Transformers (ViTs) into three main categories, addressing model compression and optimization. The first category targets model parameters and complexity, which includes: *Patch Merging* to reduce the number of operations by combining multiple input patches. *Knowledge Distillation* which involves training smaller models to replicate the performance of larger, pre-trained models. *Pruning* to remove less important parameters from the model, reducing size while maintaining performance. *Efficient Attentions* include mechanisms like sparse and low-rank attentions that minimize the computational load. *Residual Connections* which help in the training of deep networks by allowing gradients to flow directly through the layers.

The second category is Training Optimization, which includes *Batch Normalization* to target model inputs, stabilizing learning and accelerating convergence. *Gradient Accumulation* which supports training with larger batch sizes by increasing gradients over multiple steps. *Precision Management*, which depends on techniques like mixed precision and quantization-aware training to optimize computational efficiency and model accuracy. The third category is techniques targeting energy efficiency, which includes *Asynchronous Computation* to increase throughput and reduce latency by performing computations in a non-blocking manner. *Hardware Acceleration*, that uses specialized hardware to enhance the efficiency of model computations. *Offloading* techniques which transfer the computation from local (e.g., edge) devices to more powerful servers or cloud systems. *Software Acceleration* to improve software and model processing techniques to efficiently utilize hardware memory and processors; this technique further includes *Approximate Inference* to lower computational demands by approximating neural network outputs, and *Variational Parameters* to apply variational inference for optimizing parameter distribution, reducing power consumption. The figure also covers “Application Specific Optimization Approaches," which include hybrid architectures, cross-domain adaptation, and context-aware modelling techniques, which can alternatively be used or categorized within the above three categories depending on the specific architectural requirements or applications.

2.1 Efficient transformers in perception task

The efficiency of vision transformers can be categorized by their memory, parameter, and computational (processor) requirements [14, 15]. These efficiencies can be measured or correlated using performance metrics from the model and computing device [15, 16, 28]. A key factor contributing to the high memory requirement is the multi-headed self-attention mechanisms, bringing an overhead to the memory footprint of the model. The large number of parameters in the deep transformer architecture and token embeddings also further increase memory requirements during training and inference. Similarly, the intermediate results generated during the forward pass of the model, such as attention scores and feature maps, need to be stored, adding to the overall memory consumption. To tackle these challenges, computational and memory-efficient models such as EfficientViT [15] and TinyViT [2] have been proposed for optimized deployment on the edge. These models balance size and performance, reducing parameters and GFLOPs while maintaining baseline accuracy [2, 15]. Similarly, models such as LeViT and TinyCLIP show the potential to achieve computational efficiency and adaptive deployment on embedded platforms, which is essential for real-time applications [23, 34]. Another optimization approach for ViT's efficient deployment on edge involves modifying the model's computational graph and adapting it for available edge resources. This optimization method can be categorized within hardware-aware optimization for application-specific ViTs [35].

2.2 Variational inference

Variational inference (VI) provides an approach to reduce model complexity and optimizes them through balanced approximations. This approach depends on transforming intractable integration problems, often faced in calculating posterior distributions in Bayesian inference, into solvable optimization problems [36]. VI introduces a simpler, parameterized family of distributions, known as variational distributions, which are used to approximate the true posterior distributions of model parameters [32]. The objective is to find the best approximation within this family by minimizing the divergence between the variational distribution and the true posterior, measured using the Kullback–Leibler (KL) divergence.

This transformation is required to convert the problem of Bayesian integration, which is computationally expensive, into a more tractable form. By optimizing the parameters of the variational distribution, VI simplifies the model's computational requirements without trading model performance. For vision transformers, which involve complex, high-dimensional data and require substantial computational power, applying VI allows for efficient posterior approximations [33]. This method enhances the scalability of these models and supports real-time applications by reducing the computational overhead associated with traditional Bayesian methods [37]. In practice, when using VI for vision transformers, as seen in methods using energy-based priors for salient object detection, the approach involves jointly training the variational transformer with the baseline model. This training often uses

Monte Carlo-based maximum likelihood estimation, which helps fine-tune the variational parameters without requiring adversarial or complex variational learning networks [19]. The use of VI in this context shows its adaptability and effectiveness in handling the substantial computational demands of processing tasks like object detection, where maintaining a balance between model accuracy and operational efficiency is a priority [38].

Previous implementations of VI have explored in-context learning and Bayesian methods to improve the efficiency of posterior approximations. Techniques such as Prior-Data Fitted Networks utilize VI by sampling from a prior distribution within a supervised classification framework, enabling the learning of probabilistic predictions from masked label data. This approach speeds up the model convergence by following the principles of Gaussian processes, thereby enhancing the Bayesian inference process for practical applications in vision tasks [18, 19, 37, 38].

2.3 Mixed precision techniques

AI models such as CNN, DNN or ViTs are developed using several numerical components or modules such as a convolutional layer, dense layer, dropout, multi-layer perception, multi-head self-attention, feed-forward network, decoders, etc. Depending on the AI model backbone or architecture, these modules or layers can involve computations in different numerical precision (e.g., 16-bit, 32-bit, etc.) and arithmetic such as floating points or integers [11, 16, 17, 39]. Mixed precision usage in these models balances computational efficiency with model performance and has often been associated with training speed up, reducing memory requirements. Similarly, in post-model training, when the network weights or model layers are quantized with reduced precision (e.g., 32-bit to 16-bit), the gain can be seen in model inference metrics [27, 40, 41]. Such techniques are particularly beneficial in computational complex models like ViTs, which include multi-precision numerical components and modules. Multiple mixed precision training strategies, resulting in faster model convergence on single or multiple GPUs, have been previously proposed for CNN, DNN and RNN, [40, 42, 43].

For the ViTs, Wang et al. explored mixed precision techniques to optimize model deployment on edge devices by achieving a balance between power consumption and performance [44]. Post-training quantization of vision transformer, which adjusts weights and activations of the network to lower precision format after the model is trained, is combined with mixed precision formats to enable efficient deployment and inference on resource-constrained devices. In [45], the proposed method includes the integration of SmoothQuant with Bias (SQ-b) to address activation asymmetry and Optimal Scaling Factor Ratio Search (OPT-m) to automate quantization parameters while balancing model performance and model compression efficiency [45].

Another method proposed for efficient model inference includes an adaptive attention shrink module for identifying contributing patches and bit-width adjustments [46]. Within a similar scope, a mixed precision quantization using layer-wise relevance propagation has been proposed in the post-training optimization category

[47]. The proposed method improves quantization performance for ViTs by combining a novel clipped channel-wise quantization method and explainability-based mixed-precision bit allocation [47]. Another quantization strategy for faster model training and convergence includes quantization-aware training, where quantization on weights or layers is directly applied during the model training process to adjust model complexity and balance model performance metrics [44, 45]. These techniques are also used for the model fine-tuning process for specific devices or AI accelerators. These studies show that mixed precision approximation is valuable for enhancing the efficiency of complex models like ViTs, especially in scenarios where computational resources are constrained. A comparison of strategies can be seen in Table 1.

3 Method

Considering the discussed memory and computational requirements of the multi-modalities transformers, we aim to reduce the model size and computational complexity of the baseline transformer model (M) using a combination of mixed-precision quantization and variational inference. The resulting optimized model M^* (also referred to as LiteVit in the following section) will have a favourable trade-off between model performance and computational efficiency. However, to obtain the optimized model, the following challenges need to be addressed:

1. Determining optimal values for mixed-precision quantization for two modalities.
2. Assessing the impact of cross-modal bit assignment on overall model efficacy.
3. Achieving balanced metrics (e.g., energy vs accuracy).
4. Combining the optimization factors of variational inference and mixed-precision quantization.

The remainder of this section covers the proposed algorithm while discussing the above-mentioned challenges in dedicated subsections. The discussion considers “RangeVit" as the model that needs to be optimized for multi-modal data containing the properties of the nuScenes or Waymo datasets.

3.1 Overview of transformer version

We use Transfusion and RangeViT versions labelled RangeViT-v1, RangeViT-v2, and RangeViT-v3 as baseline for our training and testing setup. The versions differ primarily in channel size, affecting the number of parameters and the model’s overall complexity. RangeViT-v1, with a channel size 64, contains approximately 22.7 million parameters, positioning it as the most lightweight model among the three. RangeViT-v2 increases the channel size to 128, raising its parameter count to around 23.7 million. Lastly, RangeViT-v3, with the largest channel size 192, consists of approximately 25.2 million parameters. The purpose of using these versions is to evaluate how model size and complexity alterations influence computational

Table 1 Literature summary from quantization and mixed precision strategies for vision models

Method	Quantization Strategy	Reduction (Model)	Speed-up	Accuracy Range	Techniques
Q-ViT [48]	Hybrid (HW) Quantization	(30-40)%	6.0×	±(6-7)%	Parameter & activation efficient fine-tuning (4-bit)
QuantFormer [44]	Parameterized Quantization	(25-40)%	NA	±(2-3)%	Reconstruction-based calibration
PatchWise [46]	Hybrid Quantization	(60-80)%	2.3×	±(4-7)%	Sparsity-aware quantization
LRP-QViT [47]	Mixed Precision	(15-23)%	NA	±(7-9)%	Layer-wise adaptive (4-6 bit) quantization
MPTQ-ViT [45]	Dynamic Precision	(40-60)%	NA	±(8-12)%	Heterogeneous(4-5 bit) quantization
This Work	Dynamic Precision	(30-50)%	1.8×	±(4-5)%	VI for multi-modality + Mixed Precision QAT

Table 2 RangeVit configuration used in this paper

Model	Chan	Layers	Heads	GFLOPs	Param
v1	64	12	6	21.4	22.7M
v2	128	12	6	25.5	23.7M
v3	192	12	6	27.9	25.2M

efficiency and performance, especially in resource-limited settings like edge computing. These versions are concisely summarized in the table below, which details their specific configurations regarding channel size, layers, heads, GFLOPs, and the total number of parameters. Table 2 provides a clear overview of the architectural differences between the versions, which will be foundational in our subsequent analysis and optimization using mixed-precision quantization and variational inference. The use of these versions and configurations is driven by the hypothesis that testing and evaluating the model with varying model sizes can impact the performance and efficiency of Vision Transformers in constrained environments like edge computing. These versions will be the baseline architectures for further optimization using our proposed mixed-precision and VI algorithms.

3.2 Mixed-precision

Mixed-precision allows different model components/blocks to operate at different numerical precisions. High-precision calculations are reserved for critical parts of the model, while other parts use lower precision without significantly affecting overall performance. This strategy reduces the model's memory footprint and speeds up computation, especially beneficial for edge devices with limited memory and compute resources. In our approach, we chose the point cloud processing backbone to operate at lower precision due to its inherently large and complex nature, which demands substantial computational power. Vision Transformer pipeline handling 2D image analysis operates at higher precision to preserve the quality of visual feature extraction. This selective precision approach ensures optimal resource utilization [42, 45]. The choice of precision levels for different modules/subgroups was based on initial empirical analysis and performance evaluation, showing that the precision levels effectively balance computational load and accuracy.

1) Principles of Mixed Precision: For multi-modal models, separate precision tuning is essential. This step involves identifying distinct model components (subgroups) for each data type and tuning their precision levels independently. The optimal bit-width for each subgroup is determined by minimizing the loss function specific to its modality, as shown in Eq. 1.

$$\text{BitWidth}(S) = \underset{bw}{\text{argmin}}\, \text{Loss}(\text{Quant}(\text{M}(S), bw)) \tag{1}$$

Here, M is the model used for optimization(training), S represents a subgroup within the model, corresponding to either camera or LiDAR data, and bw represents the Bit-width used for quantization of the subgroup. Loss represents the function used

to evaluate the effectiveness of quantization. Quant represents a function that applies quantization to the weights of subgroup S at a specified bit-width bw.

2) Implementation with Modalities: In LiTeViT, the point cloud processing backbone operates at lower precision to manage computational load, while the Vision Transformer segment handling 2D image analysis operates at higher precision to preserve visual feature extraction quality. This selective precision approach ensures optimal resource utilization. Subgroup-based quantization is applied after determining the optimal bit-width for each subgroup, as shown in Eq. 2.

$$Quantized_S = \text{round}\left(\frac{\text{M}(S)}{\text{scale}_S}\right) \tag{2}$$

Here $Quantized_S$ refers to the quantized version of the weights for subgroup S. $M(S)$ represents the weights of subgroup S in the model. Subgroups could be different module/blocks (e.g., encoder) of the network that may benefit from varying levels of precision, such as layers responsible for processing point cloud data versus layers handling 2D image data. scale_S represents factor computed for subgroup S, based on the range of weights and the selected bit-width. This factor adjust the range of the weight values in subgroup S to a standard scale that aligns with the precision level determined for optimal performance. The scaling factor helps in maintaining the dynamic range of the original weights after quantization, ensuring that important information is not lost due to reduced precision.

3) Cross-Modal bit-assignment: The cross-modal bit-assignment process is designed to ensure that adjustments in the quantization of one data modality (e.g.,LiDAR) balance the metrics rather than reducing the overall model performance metrics. This is achieved through an optimization strategy which considers the joint performance metrics of both modalities (also shown in Eq. 3), which is described as following:

$$BitA = \text{Optimize}(\text{CMP}(\text{Cam}, \text{Li})) \tag{3}$$

Here BitA refers to the optimized bit assignments for the subgroups or different model components. The Optimize function strategically adjusts these bit assignments by evaluating their impact through the Cross-modal performance (CMP). This function specifically measures how changes in bit assignments during training affect the combined output quality and performance metrics of the camera and LiDAR processing pipeline, ensuring that reduction in does not affect overall model performance.

4) Loss Function: The total loss function includes multiple independent losses that balance the quantization error for both camera and LiDAR data, penalizing individual differences in performance between the two modalities. This loss function guides the model toward an optimal configuration, as shown in Eq. 4.

$$L = L_{\text{quant}}(\text{Cam}) + L_{\text{quant}}(\text{Li}) + \lambda \cdot L_{\text{reg}} \tag{4}$$

Here L is the total loss function. L_{quant} represents the quantization loss for each modality (Camera or LiDAR). λ is a hyperparameter balancing the quantization loss

and the penalty term. L_{reg} represents the penalty term for significant performance differences between the camera and LiDAR modalities. As an example of how each component of the loss function contributes to the overall model training process, ensuring a balanced optimization across different modalities and regularization, we show values logged at epoch 16 as $L_{\text{quant}}(\text{Cam}) = 0.045$, $L_{\text{quant}}(\text{Li}) = 0.038$, $L_{\text{reg}} = 0.012$, $\lambda = 0.1$. The total loss L for the above values is computed as:

$$L = 0.045 + 0.038 + 0.1 \times 0.012 = 0.0842$$

The loss values show an early phase in the training where the model starts to show improvements in handling quantization errors while maintaining a balance with regularization to prevent overfitting. This stage shows the effectiveness of the mixed-precision and quantization strategies in reducing computational complexity without higher loss in model accuracy.

Quantization, by nature, involves non-differentiable operations, posing a challenge for gradient-based optimization methods. To address this challenge Gradient approximation techniques such as the Straight-Through Estimator (STE) [39] are used. STE allows for the approximation of gradients through non-differentiable quantization functions, enabling backpropagation and thus the learning process in a mixed-precision setting. This technique is crucial for optimizing the quantized model effectively.

$$\frac{\partial L}{\partial \text{QuantWt}} = \text{STE}\left(\frac{\partial L}{\partial \text{Wt}}\right) \tag{5}$$

Here $\frac{\partial L}{\partial \text{QuantWt}}$ is the gradient of the loss function with respect to the quantized weights. *STE* represents the Straight-Through Estimator, a method used for approximating gradients through non-differentiable functions and $(\frac{\partial L}{\partial \text{Wt}})$ represents the gradient of the loss function with respect to the original (non-quantized) weights.

3.3 Variational inference

Variational inference introduces the Bayesian approach to model training. The method allows us to replace deterministic weights with distributions, transforming the model into a probabilistic one. By doing so, the model learns not just a single set of optimal weights but a distribution of possible weights, offering a robust way to gauge model uncertainty and generalize under diverse conditions.

Vision Transformer Compression: In applying variational inference to ViT, each weight of the transformer is represented as a Gaussian distribution, characterized by its mean and variance. During training, these distributions are continually updated, leading to a model that dynamically adjusts its complexity based on the certainty of its predictions. This results in a more compact and efficient model, as weights with high variance-indicating less impact on the model's output can be pruned or allocated lower precision in the mixed-precision setting.

Loss Function: The variational loss function, known as the Evidence Lower Bound (ELBO), is central to variational inference. It is formulated to maximize the

likelihood of the observed data while simultaneously regularizing the complexity of the model, thereby preventing overfitting. The ELBO balances model accuracy and compactness, a key requirement for efficient edge deployment.

$$\text{ELBO} = \mathbb{E}_{q(\theta|D)}[\log p(D|\theta)] - \text{KL}(q(\theta|D)||p(\theta)) \tag{6}$$

In this equation, $q(\theta|D)$ is the variational distribution of the model parameters θ given the data D. The first term refers to the expected log-likelihood of the data, and the second term is the Kullback–Leibler divergence between the variational distribution and the prior distribution $p(\theta)$. This formulation approximates the true posterior distribution in a computationally efficient manner.

Implementation: In Vision Transformers, variational inference is implemented by treating each weight as a variable drawn from a Gaussian distribution. During training, both the mean and variance of these distributions are optimized. This optimization is reflected in the ELBO, where the model learns to balance the fit to the data against the complexity of the weight distributions. The outcome is a model that effectively captures the uncertainty in its predictions, leading to a more robust and generalized performance. Integrating these principles with the previously discussed mixed-precision quantization strategy, we achieve a compact and efficient model in terms of computational resources and capable of maintaining high accuracy, which is essential for edge computing applications.

3.4 Algorithm for approximation using variational parameters

Our proposed algorithm integrates mixed-precision quantization and variational inference for training Vision Transformers (ViTs). This approach optimizes computational efficiency and model accuracy, particularly for data-intensive applications using large datasets such as nuScenes and Waymo. The training process begins by initializing the Vision Transformer model (M) with variational parameters (V), which allows the model to learn a probabilistic distribution of weights. This Bayesian approach enhances the model's ability to manage uncertainty and generalize across diverse conditions. The subsequent steps involve:

- **Data Processing**: The dataset (D) is divided into camera and LiDAR batches, enabling precision-specific processing. This ensures that each data modality is handled appropriately.
- **Mixed-Precision Quantization**: Different precision levels are assigned to various parts of the model based on the bit-width range (B), optimizing computational efficiency. This selective precision approach reduces computational load while maintaining predictive accuracy.
- **Loss Computation**: The training loss (L_{train}) and the variational inference loss (L_{var}) are computed. The ELBO (Evidence Lower Bound) regularizes the model's complexity. The total loss (L_{total}) is a weighted sum of these two, balancing model accuracy and complexity.

The algorithm's core includes gradient approximation techniques, such as the Straight-Through Estimator (STE), which enable effective backpropagation in mixed-precision environments. Regular validation on a separate validation set (D_{val}) and iterative updates of bit assignments further optimize performance across modalities. Upon completing the training epochs, the final model weights (ω^*) are derived based on optimized bit assignments (O^*). The model is further refined using variational inference criteria, resulting in an optimized model (M^*) suitable for real-time applications, especially in resource-constrained edge computing environments.

Algorithm 1 Training Transformers with Mixed-Precision and Variational Inference

Require: Dataset (D), Transformer-model (M), epochs (E), learning rate (η), bit-width range (B), variational parameters (V), balancing factor (λ)
Ensure: Optimized model (M^*)
  Initialize M with variational parameters V
  for $e = 1$ to E **do**
    for each *batch* in D **do**
      $C_{data}, L_{data} \leftarrow$ Split *batch* into (Camera, Lidar)
      Process C_{data} and L_{data} through M
      Apply Quantization(C_{data}, L_{data}, M) using B
      $L_{train} \leftarrow$ Compute Training Loss on D
      $L_{var} \leftarrow$ Compute Variational Inference Loss (M, V)
      $L_{total} \leftarrow L_{train} + \lambda * L_{var}$
      Apply STE for back-propagation
      Update weights ω and variational parameters V
      Update Bit Assignments for (C_{data} and L_{data})
    end for
    if $e >$ to E_{val} **then**
      Validate M on D_{val}
    end if
  end for
  Derive the final weights ω^* based on learned optimal bit assignments O^*
  Optimize M based on variational inference criteria
  $M^* \leftarrow$ Finalize M
  return M^*

Optimization for Edge: Deploying LiTeViT models on NVIDIA Jetson Xavier NX requires a comprehensive approach that implements model approximation with precision optimization for specific hardware adaptations. This subsection discusses three systematic methodologies for integrating NVIDIA's TensorRT with specific optimization techniques to enhance the efficiency of LiTeViT on the Xavier NX.

1) TensorRT Integration and Model Optimization: The LiTeViT model is first converted into a TensorRT-compatible format using NVIDIA's TensorRT API. This step involves translating the high-level model architecture into an optimized computational graph, which is deployable in the Xavier NX environment. The next step is *Precision Optimization:*. By using the TensorRT's mixed-precision approach, the model configurations are adjusted to FP16 and INT8 precision levels. This precision calibration is used for minimizing memory footprint and improving the inference

speed. The last step in this method is *Layer Fusion:*, as the TensorRT's layer fusion feature combine multiple computational layers into fewer and more efficient operations. This optimization reduces the overhead associated with memory or communication between layers, thus accelerating the computational throughput and reducing latency.

2) Advanced Quantization Techniques: Following the model conversion, another approach is to use INT8 quantization to compress the model further. This process reduce the model size, allowing for efficient utilization of the memory, which is important for maintaining high throughput on memory-constrained edge devices. Quantization-aware training can be used to ensure the robustness of the model against precision losses introduced by quantization. This technique adjusts the model parameters during the fine-tuning phase to accommodate the reduced precision, thereby preserving the accuracy post-quantization.

3) Dynamic Voltage and Frequency Scaling (DVFS): The third technique is to use the inbuilt DVFS to dynamically modulate the operating frequency and voltage of the Jetson Xavier NX based on the workload demands. This adaptive scaling manages power consumption and thermal output during compute intensive model inference tasks. The DVFS settings are aligned with the phases of model execution with TensorRT, and this synchronization ensures that the system resources are optimally allocated for energy efficiency during computationally intensive tasks. These methodologies improve the deployment of LiTeViT models, by ensuring that they meet the Jetson Xavier NX's computational constraints and deliver optimal performance in real-time applications.

4 Setup and experiments

This section covers the architecture and respective fundamental blocks of the RangeViT model, and its versions (also shown in Table 2) are explored in this paper. RangeViT [20] processes both 2D image and 3D point cloud data, making it suitable for applications such as autonomous driving. It utilizes a dual-data stream approach to handle camera and LiDAR data, incorporating the self-attention mechanism of transformers to understand complex spatial relationships. A convolutional neural network component in RangeViT projects 3D point cloud data into a 2D space, easing the computational burden while retaining essential spatial details. The fusion of 2D and transformed 3D data within RangeViT enhances its capacity to form a comprehensive understanding.

Hardware Setup: For training the RangeViT and TransFusion models, our training uses DelftBlue supercomputer configurations. As the nuScenes dataset has a high memory requirement, we use a computing configuration which includes Intel Xeon high-performance multi-core processor with NVIDIA GPUs V100. The setup uses 4 GPUs with 16 GB of VRAM to handle large model sizes and data processing requirements. High-end system memory is required with the configuration to accommodate the nuScenes and Waymo datasets and to facilitate efficient data loading or input feature maps. This configuration ensures sufficient computational power

(a) nuScenes[29] (b) Waymo[30]

Fig. 3 Sample frame from the dataset

for training these complex vision transformer models for segmentation and detection using LiDAR-camera fusion for vision applications.

Dataset and Preprocessing: The nuScenes dataset [29], is a comprehensive dataset designed for autonomous driving applications (sample frames are shown in Fig. 3a). It provides a diverse range of scenarios captured in urban environments, encompassing both camera images and LiDAR point clouds. It consists of 1,000 scenes captured across Singapore and Boston (USA). These scenes are of around 20 s. It further consists of 16 semantic classes. To train our model, we follow the RangeVit method of data preprocessing, modality handling and splitting 28,130 training and 6,019 validation point cloud scans [20].

Training with Algorithm: The training approach is used to optimize the models for edge computing, focusing on efficiency and model complexity balance. We use the Adam optimizer for training, with settings of $\beta 1 = 0.9$, $\beta 2 = 0.999$, and a weight decay of 0.0001, with a batch size of 32. We start with 0.1 and increment the learning rate using cosine decay in every sequence of epochs. This approach ensures a comprehensive optimization of the models. Post-training, each RangeViT (models with different configurations) is adapted or modified as a LiteViT version with reduced model size, balanced computational efficiency, and performance. These LiteViT models are then evaluated against accuracy, latency, and energy efficiency benchmarks, affirming their effectiveness and adaptability for real-world edge computing applications. The balanced optimization of RangeViT models into more efficient LiteViT versions shows the efficiency of the proposed training algorithm, highlighting its potential in the model acceleration of vision transformers for edge deployment scenarios.

5 Results and evaluation

After obtaining the LiteVit models, we compare the results with models that have been benchmarked before on nuScenes and are within the same range of parameters, flops, and performance.

Model Evaluation: We observed a variation in performance among different models when tested on the nuScenes dataset to capture mIoU, Latency and GFLOPs. Table 3 presents these metrics in detail. With fewer parameters and lower GFLOPs,

Table 3 Trained model evaluation on the nuScenes validation set using computation and model performance metrics (Flops, Latency and mIoU)

Model	#Para	GFLOPs	Latency (ms)	mIoU(%)
TransFusion	27.8M	38.3	268.2	69.3
RangeViT-1	21.4M	35.4	188.2	65.1
RangeViT-2	25.5M	38.6	212.5	69.0
RangeViT-3	27.9M	39.4	236.9	74.6
LiTeViT-1	10.3M	24.4	122.9	72.9
LiTeViT-2	16.7M	32.1	145.1	73.8
LiTeViT-3	19.1M	33.7	155.2	74.6

LiTeViT model versions outperform or match the more computationally intensive TransFusion and RangeViT models in mean Intersection over Union (mIoU), while having lower latency. For instance, LiTeViT-1, with just 10.3M parameters, achieves a competitive mIoU of 72.9% at a latency of 122.9 ms, showcasing its suitability for resource-constrained scenarios without compromising performance.

We also evaluate the trained model (three-versions) performance on the Waymo validation dataset (sample shown in Fig. 3b) using object detection task, specifically for the vehicles (car) and pedestrians class. The evaluation metrics, as shown in Table 4a, present Average Precision (AP) and Average Precision with Heading (APH) for both vehicle and pedestrian categories. In evaluation, RangeViT models show high performance, with RangeViT-v3 reaching AP scores of 69.4 for vehicles and 69.9 for pedestrians. LiteViT versions show balanced efficiency and accuracy, with LiTeViT-v3 having an Average Precision (AP) score of 67.4 for vehicles and 72.4 for pedestrians, alongside Average Precision with Heading (APH) scores of 67.1 and 70.0, respectively. These results show the LiteViT models' adeptness in balancing computational efficiency with competitive accuracy, showcasing their potential for real-world application.

Energy Evaluation: Table 4b compares various models tested on the nuScenes validation set. The LiteViT versions, particularly LiTeViT-1 and LiTeViT-3, show a balance between accuracy and energy consumption. LiTeViT-1, consuming around 121.0 mJ, attains a mean Average Precision (mAP) of 70.5%, while LiTeViT-3, with an energy usage of 379.6 mJ, shows an mAP of 78.3% among tested models. A direct comparison can be seen with the energy-intensive TransFusion model, which shows an mAP of 66.8% and requires around 507.4 mJ for processing the same batch size. The RangeViT series shows performance improvements in mAP, resulting in RangeViT-3's 78.3% mAP at an energy cost of 502.9 MJ (these power metrics were measured using nvpmodel GUI). The approach and results discussed for variational inference use a Bayesian methodology in model training, transforming deterministic weights into probabilistic distributions. To observe the potential and benefits of the training algorithm, this subsection explores the implementation of the Laplace distribution as an alternative to the Gaussian distribution within transformers, analyzing its impact on model performance and efficiency.

Table 4 Results on Waymo and nuScenes dataset with a batch size of 32

	Vehicle		Pedestrian	
Model	AP	APH	AP	APH
(a) Results on the Waymo open dataset				
Transfusion	58.4	57.9	56.8	54.3
FusionViT	59.5	58.4	54.6	54.8
RangeViT-v1	64.4	63.7	68.9	57.2
RangeViT-v2	68.1	68.2	71.4	66.0
RangeViT-v3	69.4	69.1	69.9	66.8
LiTeViT-v1	61.3	60.6	65.5	64.1
LiTeViT-v2	63.3	62.9	68.8	56.4
LiTeViT-v3	67.4	67.1	72.4	70.0
Model	mAP	Car	Ped	E(mJ)
(b) Energy(mJ) consumption for a batch				
TransFusion	66.8	74.1	59.5	507.4
BEVFusion	67.4	71.9	62.9	490.2
VPFNet	62.9	67.1	58.7	495.7
RangeViT-1	74.8	80.3	69.3	416.2
RangeViT-2	75.9	81.7	70.1	480.1
RangeViT-3	78.3	86.2	70.4	502.9
LiTeViT-1	70.5	78.2	62.7	121.0
LiTeViT-2	71.2	76.9	65.5	194.6
LiTeViT-3	78.3	81.8	74.6	379.6

5.1 Exploring the laplace distribution

Characterized by its mean μ and scale parameter b, the Laplace distribution offers a unique profile for weight distributions with heavier tails. The probability density function (PDF) for the Laplace distribution in our case is defined as:

$$f(x|\mu, b) = \frac{1}{2b} \exp\left(-\frac{|x-\mu|}{b}\right).$$

In our transformer model, weights are represented as variables drawn from this distribution. The optimization process adjusts μ and b to maximize model fit while regularizing its complexity, as encapsulated by the Evidence Lower Bound:

$$\text{ELBO}_{\text{Laplace}} = \mathbb{E}_{q(w|D)}[\log p(D|w)] - \text{KL}(q(w|D)||p(w)),$$

where $q(w|D)$ is the variational distribution of model parameters w given the data D, and $p(w)$ is the prior distribution.

Comparative Analysis: We compare the Laplace distribution against models using Gaussian variational inference across metrics such as model accuracy, GFLOPs, and latency in Table 5. The analysis indicates that using a Laplace

Table 5 Comparative results using Gaussian and Laplace distributions for Variational Inference

Model	Distribution	#Params	GFLOPs	Latency (ms)	mIoU (%)
TransFusion	Gaussian	27.8M	38.3	268.2	69.3
LiTeViT-1	Laplace	10.3M	23.0	120.5	72.4
LiTeViT-2	Laplace	16.7M	30.7	142.8	73.5
LiTeViT-3	Laplace	19.1M	31.9	150.0	74.1

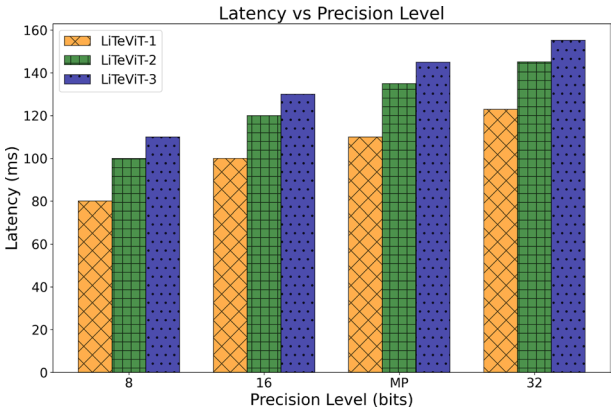

Fig. 4 Latency

distribution for variational inference enables the LiTeViT models to maintain competitive mIoU percentages while achieving reductions in GFLOPs and latency. These results suggest an effective optimization for resource-constrained edge computing scenarios, balancing performance with computational demand.

In analyzing latency as a function of precision level (Fig. 4), the models shows an inverse relationship. Higher precision configuration results in increased latency, while lower precision improves processing speed. This pattern shows the advantages of precision scaling in critical real-time applications, such as those in autonomous vehicles where rapid data processing is essential. LiTeViT-1 and LiTeViT-2 show improvements in reducing latency, especially at 8-bit precision, reinforcing the practicality of using lower precision levels for non-critical components of the models. The mixed precision configuration can be considered adaptable to varying computational demands, ensuring that the models maintain acceptable performance thresholds without unnecessary delays. While these tests are run on general GPU devices, as discussed earlier, the potential of proposed techniques can also be explored with custom accelerators designed for application-specific tasks.

The memory usage of models in Fig. 5 shows the impact of precision adjustments on system resource allocations. As expected from theory, the lower precision levels reduce the memory footprint, with LiTeViT-1 and LiTeViT-3 at 8 bits. This reduction is necessary for edge devices typically constrained by onboard memory (RAM) resources. By using lower precision, these models can handle larger datasets or run

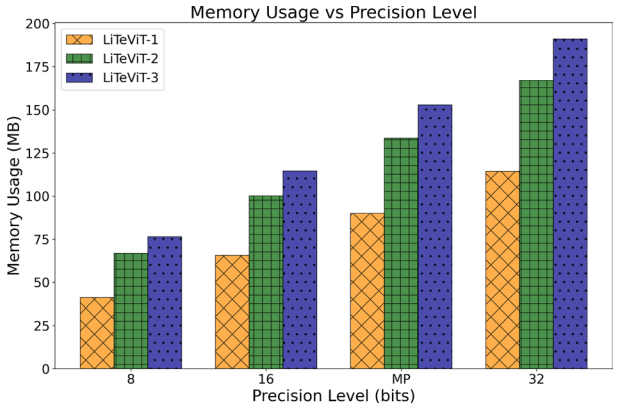

Fig. 5 Memory

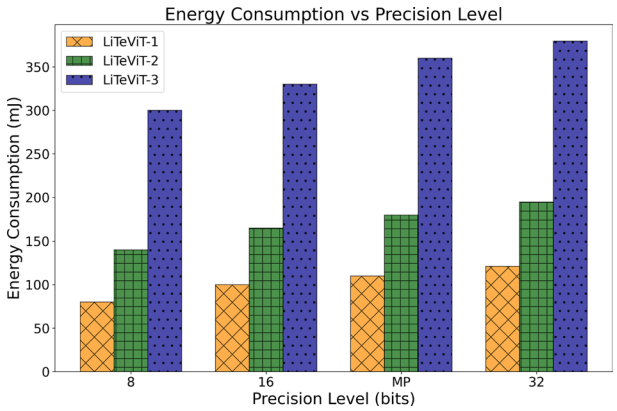

Fig. 6 Energy Comparison

multiple applications simultaneously without significantly compromising overall performance. The mixed precision approach further improves this advantage by selectively applying precision levels, thus optimizing memory usage without sacrificing essential model accuracy. The combination improves operational efficiency and extends these models' applicability to various real-time and memory-sensitive environments.

Figure 6 compares energy consumption and precision level across the three versions of the LiteViT model. As the precision decreases from 32 bits to 8 bits, the energy required by the models drops, with the most decrease observed in LiTeViT-1. This trend is consistent across all models, showing the efficiency of adopting lower precision bits. The implementation of mixed precision (MP) shows a strategic balancing of lower energy demands while still keeping higher precision in the model's critical components (i.e., affecting accuracy). Overall, these observations confirm that strategic precision scaling across model components can lead to improvements

Table 6 Quantization Strategy and Performance Outcomes (MP = Mixed Precision)

Model Configuration	Precision (Bits)	GFLOPs	Latency (ms)	mIoU (%)
Baseline (Full Precision)	32	39.4	268.2	79.7
LiTeViT-1 (MP A)	Encoder: 16, Others: FP8	20.3	100.7	72.4
LiTeViT-2 (MP B)	Encoder: 16, Others: Int8	28.9	118.3	73.1
LiTeViT-3 (MP C)	FP 16	29.5	127.5	73.9

in energy efficiency, latency, and memory usage, which are essential for the deployment of deep learning models in resource-constrained edge devices.

5.2 Exploration of quantization depth

Building on our proposed algorithm (Section 3.4) for training transformers with mixed-precision and variational inference, this subsection explores the "Quantization" step, quantitatively evaluating its impact on computational efficiency and model accuracy. The "Quantization" is applied for adjusting bit-width assignments across high-precision model components (e.g., FFN and MHSA) to reduce the computational bound without significantly affecting model performance. This process is necessary as it enables the precision levels to enhance model efficiency on dedicated hardware.

Comparative Analysis: We present a comparative analysis of the LiteViT models by showing the mixed-precision quantization configurations, a mix of floating points and integers in Table 6. The table shows GFLOPs and latency measurements for configurations with varying precision levels against the baseline models to highlight the improvements.

This analysis shows the efficiency gains for **LiTeViT-1** under the Mixed Precision A configuration, which shows a significant reduction in GFLOPs and latency, attesting to the approach's effectiveness in optimizing computational demands while preserving a high level of accuracy. The evaluation of our algorithm's "Apply Quantization" step highlights its role in achieving an optimal balance between computational efficiency and model performance. Future work will explore the automation of precision-level assignments and the potential integration of mixed-precision quantization with other model optimization techniques to enhance the deployment of Vision Transformers in more memory and compute-constrained environments.

Evaluation of the proposed training approach shows improvements in inference speedup and energy efficiency across all three versions of the LiTeViT model. The results, also covered in Table 7, show that each version of the LiTeViT model improves speed up and energy saving. The first row is used as a baseline or reference for comparison (configuration mentioned in Table 5). The second and third row shows results by varying the batch size from 32 to 64 respectively. Among the three versions, the best performance is seen in the LiTeViT-3 for speedup and energy savings. As seen in the table, the processing speed and energy saving decrease with the data size (batch) increase, which can be further optimized using data processing

Table 7 Results for Inference (a batch) Speedup and Energy Saving on models

Speedup (×)			Energy Saving (×)		
LiTeViT-1	LiTeViT-2	LiTeViT-3	LiTeViT-1	LiTeViT-2	LiTeViT-3
1.00	1.00	1.00	1.00	1.00	1.00
1.84	1.91	1.95	1.72	1.83	1.90
1.29	1.15	1.61	1.21	1.22	1.58

mechanisms in the model backbone. The overall results in this table further show the effectiveness of applied optimizations, positioning these models as viable solutions for performance-critical and resource-constrained environments.

5.3 Summary and key takeaways

The work covers efficient training and inference strategies for high-volume data-dependent vision transformer models. The key exploration and learning can be described as:

1. Performance Balance: The “lite" versions achieve a balanced trade-off between computational efficiency and accuracy. Utilizing our proposed hybrid training method, these models maintain robust performance with reduced parameter, highlighting their suitability for environments with limited computational resources.
2. Energy Efficiency: The on-device energy evaluations show the capability of achieving high accuracy while conserving power. LiTeViT-3 model shows the highest accuracy, whereas LiTeViT-2 offers an optimal balance between energy consumption and accuracy, making it ideal for sustained deployment in energy-sensitive applications.
3. Quantization: By implementing mixed precision strategies, the LiTeViT models show effective reductions in computational demand and energy usage without a significant drop in model performance. Particularly, LiTeViT-2 and LiTeViT-3 during increased shows speedup and savings compared to the baseline.

6 Conclusion

This paper proposes an optimization approach for ViTs to facilitate their deployment within the edge environment by exploring mixed-precision quantization combined with variational inference. The paper addresses the challenge of high-performance computational requirements arising from complex layers/components in the ViTs. These complexities make them computationally expensive and energy-demanding, a major challenge for edge deployment. By integrating variational inference and mixed-precision quantization, our training and approximation methodology reduce computational resource requirements and maintain the accuracy levels necessary

for practical applications. To explore the balance between model accuracy and performance metrics, we evaluate our proposed method on the nuScenes and Waymo datasets, showing practical optimization strategies of AI models for edge computing. While recognizing the challenges of environmental variability and the need for diverse training datasets, our work can be considered as a step for future research to enhance model adaptability and efficiency in real-world scenarios. Progress in this research area can lead to optimized AI models that are adaptive towards arithmetic and reduced precision, thus aligning the concept of limited computational resources, model performance and the need for sustainable solutions.

Acknowledgements This work was supported by the European Union's Horizon 2020 Research and Innovation Programme, under the Marie Skłodowska Curie grant agreement No. 956090 (APROPOS), and from the Flemish Government under the "Onderzoeksprogramma Artificiele Intelligentie (AI) Vlaanderen" programme.

Author contributions All authors contributed equally on methodology, writing and review process.

Data availability No datasets were generated or analysed during the current study.

Declarations

Competing interests The authors declare no competing interests.

References

1. Han K, Wang Y, Chen H, Chen X, Guo J, Liu Z, Tang Y, Xiao A, Xu C, Xu Y et al (2022) A survey on vision transformer. IEEE Trans Pattern Anal Mach Intell 45(1):87–110
2. Wu K, Zhang J, Peng H, Liu M, Xiao B, Fu J, Yuan L (2022) Tinyvit: Fast pretraining distillation for small vision transformers. In: European Conference on Computer Vision, pp. 68–85. Springer
3. Lu Z, Xie H, Liu C, Zhang Y (2022) Bridging the gap between vision transformers and convolutional neural networks on small datasets. Adv Neural Inf Process Syst 35:14663–14677
4. Ma D, Zhao P, Jiao X (2023) Perfhd: Efficient vit architecture performance ranking using hyperdimensional computing. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 2229–2236
5. Li X, Ding H, Yuan H, Zhang W, Pang J, Cheng G, Chen K, Liu Z, Loy CC (2024) Transformer-based visual segmentation: A survey. IEEE Transactions on Pattern Analysis and Machine Intelligence
6. Xiang X, Zhang J (2023) Fusionvit: Hierarchical 3d object detection via lidar-camera vision transformer fusion. arXiv preprint arXiv:2311.03620
7. Chen RJ, Lu MY, Weng W-H, Chen TY, Williamson DF, Manz T, Shady M, Mahmood F (2021) Multimodal co-attention transformer for survival prediction in gigapixel whole slide images. In: Proceedings of the IEEE/CVF International Conference on Computer Vision, pp. 4015–4025

8. Bai X, Hu Z, Zhu X, Huang Q, Chen Y, Fu H, Tai C-L (2022) Transfusion: Robust lidar-camera fusion for 3d object detection with transformers. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 1090–1099
9. Liu Y, Chen S, Lei Z, Wang P (2023) Energy consumption optimization of swin transformer based on local aggregation and group-wise transformation. In: 2023 4th International Conference on Computer Vision, Image and Deep Learning (CVIDL), pp. 463–471. IEEE
10. Mehta S, Rastegari M (2022) MobileViT: Light-weight, General-purpose, and Mobile-friendly Vision Transformer
11. Katare D, Ding AY (2023) Energy-efficient edge approximation for connected vehicular services. In: 2023 57th Annual Conference on Information Sciences and Systems (CISS), pp. 1–6. IEEE
12. Patterson D, Gilbert JM, Gruteser M, Robles E, Sekar K, Wei Y, Zhu T (2024) Energy and emissions of machine learning on smartphones vs. the cloud. Commun ACM 67(2):86–97
13. Patterson D, Gonzalez J, Hölzle U, Le Q, Liang C, Munguia L-M, Rothchild D, So DR, Texier M, Dean J (2022) The carbon footprint of machine learning training will plateau, then shrink. Computer 55:18–28
14. Shuvo MMH, Islam SK, Cheng J, Morshed BI (2022) Efficient acceleration of deep learning inference on resource-constrained edge devices: A review. Proceedings of the IEEE
15. Liu X, Peng H, Zheng N, Yang Y, Hu H, Yuan Y (2023) Efficientvit: Memory efficient vision transformer with cascaded group attention. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 14420–14430
16. Tang Y, Wang Y, Guo J, Tu Z, Han K, Hu H, Tao D (2024) A survey on transformer compression. arXiv e-prints, 2402
17. Katare D, Perino D, Nurmi J, Warnier M, Janssen M, Ding AY (2023) A survey on approximate edge ai for energy efficient autonomous driving services. IEEE Communications Surveys & Tutorials
18. Sun W, Paiva AR, Xu P, Sundaram A, Braatz RD (2020) Fault detection and identification using bayesian recurrent neural networks. Comput Chem Eng 141:106991
19. Kim H-R, Kim K-J, Choi D-H (2022) Domain adaptive detector via variational inference. In: 2022 International Conference on Platform Technology and Service (PlatCon), pp. 86–91. IEEE
20. Ando A, Gidaris S, Bursuc A, Puy G, Boulch A, Marlet R (2023) Rangevit: Towards vision transformers for 3d semantic segmentation in autonomous driving. In: CVPR
21. Dosovitskiy A, Beyer L, Kolesnikov A, Weissenborn D, Zhai X, Unterthiner T, Dehghani M, Minderer M, Heigold G, Gelly S, et al (2020) An image is worth 16x16 words: Transformers for image recognition at scale. arXiv preprint arXiv:2010.11929
22. Han K, Xiao A, Wu E, Guo J, Xu C, Wang Y (2021) Transformer in transformer. Adv Neural Inf Process Syst 34:15908–15919
23. Wu K, Peng H, Zhou Z, Xiao B, Liu M, Yuan L, Xuan H, Valenzuela M, Chen XS, Wang X, *et al* (2023) Tinyclip: Clip distillation via affinity mimicking and weight inheritance. In: Proceedings of the IEEE/CVF International Conference on Computer Vision, pp. 21970–21980
24. Carion N, Massa F, Synnaeve G, Usunier N, Kirillov A, Zagoruyko S (2020) End-to-end object detection with transformers. In: European Conference on Computer Vision, pp. 213–229. Springer
25. Yu X, Tang L, Rao Y, Huang T, Zhou J, Lu J (2022) Point-bert: Pre-training 3d point cloud transformers with masked point modeling. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 19313–19322
26. Zhao H, Jiang L, Jia J, Torr PHS, Koltun V (2021) Point transformer. In: Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV), pp. 16259–16268
27. Wang Y, Ye T, Cao L, Huang W, Sun F, He F, Tao D (2022) Bridged transformer for vision and point cloud 3d object detection. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition
28. Papa L, Russo P, Amerini I, Zhou L (2024) A survey on efficient vision transformers: algorithms, techniques, and performance benchmarking. IEEE Transactions on Pattern Analysis and Machine Intelligence
29. Caesar H, Bankiti V, Lang AH, Vora S, Liong VE, Xu Q, Krishnan A, Pan Y, Baldan G, Beijbom O (2020) nuscenes: A multimodal dataset for autonomous driving. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 11621–11631
30. Sun P, Kretzschmar H, Dotiwalla X, Chouard A, Patnaik V, Tsui P, Guo J, Zhou Y, Chai Y, Caine B, *et al* (2020) Scalability in perception for autonomous driving: Waymo open dataset. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 2446–2454

31. Team OD (2020) OpenPCDet: An Open-source Toolbox for 3D Object Detection from Point Clouds. https://github.com/open-mmlab/OpenPCDet
32. Park J, Haran M (2018) Bayesian inference in the presence of intractable normalizing functions. J Am Stat Assoc 113(523):1372–1390
33. Zhang J, Xie J, Barnes N, Li P (2021) Learning generative vision transformer with energy-based latent space for saliency prediction. Adv Neural Inf Process Syst 34:15448–15463
34. Graham B, El-Nouby A, Touvron H, Stock P, Joulin A, Jégou H, Douze M (2021) Levit: a vision transformer in convnet's clothing for faster inference. In: Proceedings of the IEEE/CVF International Conference on Computer Vision, pp. 12259–12269
35. Reidy BC, Mohammadi M, Elbtity ME, Z R (2023) Efficient deployment of transformer models on edge tpu accelerators: A real system evaluation. In: Architecture and System Support for Transformer Models (ASSYST@ ISCA 2023)
36. Greff K, Kaufman RL, Kabra R, Watters N, Burgess C, Zoran D, Matthey L, Botvinick M, Lerchner A (2019) Multi-object representation learning with iterative variational inference. In: International Conference on Machine Learning, pp. 2424–2433. PMLR
37. Müller S, Hollmann N, Pineda-Arango S, Grabocka J, Hutter F (2021) Transformers can do bayesian inference. CoRR **abs/2112.10510**
38. Zhang Y, He H, Li J, Li Y, See J, Lin W (2021) Variational pedestrian detection. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), pp. 11622–11631
39. Yin P, Lyu J, Zhang S, Osher S, Qi Y, Xin J (2019) Understanding straight-through estimator in training activation quantized neural nets. arXiv preprint arXiv:1903.05662
40. Carmichael Z, Langroudi HF, Khazanov C, Lillie J, Gustafson JL, Kudithipudi D (2019) Performance-efficiency trade-off of low-precision numerical formats in deep neural networks. In: Proceedings of the Conference for Next Generation Arithmetic 2019, pp. 1–9
41. Wang H, Xie S, Lin L, Iwamoto Y, Han X-H, Chen Y-W, Tong R (2022) Mixed transformer u-net for medical image segmentation. In: ICASSP 2022-2022 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), pp. 2390–2394. IEEE
42. Micikevicius P, Narang S, Alben J, Diamos G, Elsen E, Garcia D, Ginsburg B, Houston M, Kuchaiev O, Venkatesh G, Wu H (2018) Mixed precision training. In: International Conference on Learning Representations. https://openreview.net/forum?id=r1gs9JgRZ
43. Jia X, Song S, He W, Wang Y, Rong H, Zhou F, Xie L, Guo Z, Yang Y, Yu L, et al (2018) Highly scalable deep learning training system with mixed-precision: Training imagenet in four minutes. arXiv preprint arXiv:1807.11205
44. Wang Z, Wang C, Xu X, Zhou J, Lu J (2022) Quantformer: Learning extremely low-precision vision transformers. IEEE Transactions on Pattern Analysis and Machine Intelligence
45. Tai YS, Wu A-Y (2024) Mptq-vit: Mixed-precision post-training quantization for vision transformer. ArXiv **abs/2401.14895**
46. Xiao J, Li Z, Yang L, Gu Q (2023) Patch-wise mixed-precision quantization of vision transformer. In: 2023 International Joint Conference on Neural Networks (IJCNN), pp. 1–7. IEEE
47. Ranjan N, Savakis A (2024) Lrp-qvit: Mixed-precision vision transformer quantization via layer-wise relevance propagation. arXiv preprint arXiv:2401.11243
48. Li Y, Xu S, Zhang B, Cao X, Gao P, Guo G (2022) Q-vit: accurate and fully quantized low-bit vision transformer. Adv Neural Inf Process Syst 35:34451–34463