EnViT: Enhancing the Performance of Early-Exit Vision Transformers via Exit-Aware Structured Dropout-Enabled Self-Distillation

Yonghao Dong[1,2,3,4], Qiang He[1,2,3,4,5*], Penghong Rui[1,2,3,4], Zhenzhe Zheng[6], Zhao Li[7], Feifei Chen[8], Hai Jin[1,2,3,4], Yun Yang[5]

[1]National Engineering Research Center for Big Data Technology and System, Wuhan, China
[2]Services Computing Technology and System Lab, Wuhan, China
[3]Cluster and Grid Computing Lab, Wuhan, China
[4]School of Computer Science and Technology, Huazhong University of Science and Technology, Wuhan, China
[5]Swinburne University of Technology, Melbourne, Australia
[6]Shanghai Jiao Tong University, Shanghai, China
[7]Zhejiang Lab, Hangzhou, China
[8]Deakin University, Melbourne, Australia
{tung, hqiang, raypennhugh, hjin}@hust.edu.cn, zhengzhenzhe@sjtu.edu.cn, lzjoey@gmail.com, feifei.chen@deakin.edu.au, yyang@swin.edu.au

Abstract

Vision Transformers (ViTs) have gained significant attention and widespread adoption due to their impressive performance in various computer vision tasks. However, in practice, their substantial computational overhead often leads to high inference latency and increased overheads when deployed on resource-constrained edge devices like smartphones, autonomous vehicles, and robots. To address these challenges, Early Exit (EE) has emerged as a promising approach for lightweight inference on edge devices. It accelerates inference and reduces computational overhead by adaptively producing predictions through early exits based on sample complexity. Existing EE methods typically suffer from substantial accuracy decreases in late exits while providing only marginal accuracy improvements to early exits. This paper presents EnViT, an exit-aware structured dropout-enabled self-distillation approach that enhances the performance of early exits without compromising late exits. EnViT leverages structured dropout to enable self-distillation, where the full model serves as the teacher and its own virtual sub-models generated by structured dropout as students. This mechanism effectively distills knowledge from the full model to early exits and avoids performance degradation in late exits by mitigating parameter conflicts across exits during training. Evaluation on five datasets shows that our EnViT achieves accuracy improvements ranging from 0.36% to 7.92% while maintaining competitive speed-ups of 1.72x to 2.23x.

1 Introduction

In recent years, Vision Transformers (ViTs) (Dosovitskiy et al. 2021) have emerged as revolutionary architectures with remarkable performance across diverse computer vision tasks. ViTs leverage the self-attention mechanism (Vaswani et al. 2017) to capture long-range dependencies and global context information. Their outstanding capabilities have led to increasing adoption in academic research and real-world applications that deploy ViTs on edge devices like smartphones and drones to enable real-time services (Mehta and Rastegari 2022; Xu et al. 2025a).

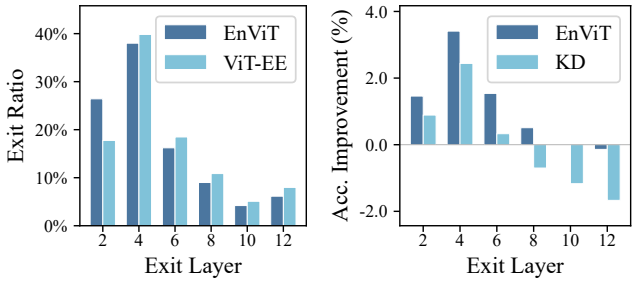
Model	Params.	FLOPs	Latency
ViT-Base	86M	17.6 GFLOPs	56ms
ViT-Large	307M	61.6 GFLOPs	154ms
ViT-Huge	632M	167.3 GFLOPs	411ms

Table 1: Inference overhead of ViTs on Jetson Orin NX.

However, the computational intensity of ViT models presents significant challenges when deployed on resource-constrained edge devices. As illustrated in Table 1, even the smallest ViT-Base model with 86M parameters requires 17.6 GFLOPs and incurs a 56ms latency to classify an image on an edge device powered by NVIDIA Jetson Orin NX. More accurate ViT-Large and ViT-Huge models exhibit even higher latencies, reaching 154ms and 411ms, respectively. This substantial computational overhead translates to high inference latency, increased energy consumption, thermal throttling, and eventually compromises user experience. Consequently, accelerating inference while reducing computational overhead is crucial for ViT deployments on these edge devices.

A variety of techniques have been proposed to create efficient ViTs, including knowledge distillation (Hinton, Vinyals, and Dean 2015; Wu et al. 2022), network pruning (Han, Mao, and Dally 2015; Zheng et al. 2022), model quantization (Han, Mao, and Dally 2015; Liu et al. 2021), and neural architecture search (NAS) (Chen et al. 2021). These techniques primarily accelerate model inference by reducing model size. However, they overlook the varying complexities of real-world data. All samples, easy or difficult, follow the same inference path, leading to overthinking issues (Zhu 2021; Bajpai and Hanawal 2025), i.e., the model expends unnecessary computation on simple samples that could be resolved with fewer layers.

*Corresponding author.



(a) Exit Ratio Comparison (b) Accuracy Comparison

Figure 1: Motivation for EnViT: (a) Sample exit ratio across different exit layers on the CIFAR-100 dataset with the same overall accuracy. ViT-EE simply adds a classification head to exit layers. (b) Comparison of accuracy improvement between vanilla knowledge distillation and EnViT. The baseline is ViT-EE.

In contrast, adaptive inference emerges as a promising technique that dynamically adjusts the computational paths of inputs based on their complexities (Rao et al. 2021; Yin et al. 2022; Zhou et al. 2020; Xu et al. 2023). It handles simple samples with minimal computational overhead and allocate more resources for complex samples. It also enables flexible control over speed-up ratio by adjusting hyperparameters like thresholds.

Early exits (EE) (Teerapittayanon, McDanel, and Kung 2016) is one of the most important methods for implementing adaptive inference. It adds exits at the intermediate layers of a model. When certain exit conditions are met, samples are allowed to exit early without passing through the entire model. This allows simple samples to be processed by early layers, and only complex samples proceed to late layers for further processing. Early exit has demonstrated promising performance across various model architectures, including CNNs (Teerapittayanon, McDanel, and Kung 2016; Huang et al. 2018) and transformer-based models (Zhou et al. 2020; Xu et al. 2023; Bajpai and Hanawal 2025). We identify the challenges of implementing effective early exit from two perspectives.

First, Limited Performance of Early Exits. The performance of early exits is a critical factor that directly determines the efficiency of early exit models, as it influences the proportion of samples that exit at intermediate layers. However, early exits face an inherent limitation: they must make decisions based on lower-level features (Chefer, Gur, and Wolf 2021; Sajjad et al. 2023), which inevitably compromises their performance. Figure 1a demonstrates the impact of early exit performance on the proportion of samples exiting at each exit. Compared to the naive approach, EnViT effectively enhances the performance of early exits. The number of samples exiting at the first exit increases by 1.5x, thereby improving the speed-up ratio by 10%. This result validates that strengthening early exit performance allows more samples to exit at early layers while preserving prediction accuracy, thereby achieving superior efficiency gains.

Second, Accuracy Degradation of Late Exits. Knowledge distillation has been widely adopted to overcome the performance limitations of early exits by transferring high-level features from late exit to early exits (Hinton, Vinyals, and Dean 2015; Liu et al. 2020; Xu et al. 2023). However, as illustrated in Figure 1b, incorporating knowledge distillation during training leads to accuracy degradation in late exits while yielding only marginal accuracy improvements in early exits. The accuracy of the last three exits decreases by -0.70%, -1.17%, and -1.67%, respectively. This degradation is particularly problematic as it compromises the model's ability to handle difficult samples. Early exit models are typically trained with a loss function that ensembles the losses from all exits into a unified objective function (Teerapittayanon, McDanel, and Kung 2016). The underlying cause stems from the additional gradient propagation paths introduced by the distillation losses, which exacerbate the inherent gradient conflicts in multi-exit training (Zhu et al. 2021; Gong et al. 2024).

This paper presents EnViT, an exit-aware structured dropout-enabled self-distillation training approach that enhances the accuracy of early exits while preserving the performance of late exits. To address the gradient conflict issues in multi-exit training, we introduce an exit-aware structured dropout training methodology. As shown in Figure 2, EnViT can be viewed as dynamically creating multiple virtual sub-models during training. Different samples pass through their respective virtual sub-models, where each sub-model serves as a student to receive knowledge from the final exit. During backpropagation, these students are ensembled together to update shared parameters. This approach effectively mitigates the gradient conflicts between different samples in the same batch across various exits. We assign higher dropout probabilities to deeper layers to encourage more samples to exit early.

Contributions. The main contributions of this paper are threefold:

- We disclose the key factors affecting early exit models from two critical perspectives: the performance bottleneck of early exits that determines model efficiency, and the accuracy degradation of late exits that compromises model performance.
- We propose a novel exit-aware structured dropout-enabled self-distillation training method termed EnViT that simultaneously enhances early exits while preserving the performance of late exits.
- We conduct comprehensive experiments to verify the effectiveness of EnViT across five datasets. EnViT outperforms state-of-the-art early exit methods, achieving accuracy improvements ranging from 0.36% to 7.92% while maintaining competitive speed-ups of 1.72x to 2.23x.

2 Related Work

Adaptive Inference. Adaptive inference enables DNN models to dynamically adjust their computation based on inputs and application scenarios (Xu et al. 2025b). For example, ACT (Graves 2016) enables a DNN model to adaptively determine the number of computational steps per input. SkipNet (Wang et al. 2018) learns to dynamically skip network

layers during inference to reduce computational cost. With the popularity of Transformer (Vaswani et al. 2017), corresponding adaptive inference techniques have emerged. DynamicViT (Rao et al. 2021) progressively reduces redundant patches based in input-dependent important scores. AdaViT (Meng et al. 2022) learns to dynamically select which patches, attention heads, and Transformer blocks to use on a per-input basis. SP-ViT (Kong et al. 2022) uses an attention-based multi-head token selector and token packaging technique to achieve per-image computation adaptation. A-ViT (Yin et al. 2022) dynamically halts tokens at different depths based on their importance through a parameter-free halting mechanism that reuses network parameters. PuMer (Cao, Paranjape, and Hajishirzi 2023) uses text-informed pruning and modality-aware merging to reduce tokens.

Early Exit. Early exit is an adaptive inference strategy that incorporates exit points at the intermediate layers of a model (Teerapittayanon, McDanel, and Kung 2016). It enables samples to exit early from the model based on exiting conditions like confidence or resource constraints. Studies of early exit can be categorized into two main classes, i.e., model designs and training schemes.

- **Model Designs.** MSDNet (Huang et al. 2018) introduces a multi-scale dense network architecture that enables efficient anytime prediction by maintaining coarse-level features and using dense connectivity to support early-exit classifiers. PABEE (Zhou et al. 2020) adds classifiers at intermediate layers so that the model can stop computation when consecutive predictions stabilize for a number of steps. LGViT (Xu et al. 2023) incorporates local perception heads for early layers and global aggregation heads for deep layer along with a two-stage training strategy to achieve efficient dynamic inference.
- **Training Schemes.** DFS (Gong et al. 2024) addresses gradient conflicts in multi-exit networks through feature partitioning along the depth axis and feature referencing across exits. LayerSkip (Elhoushi et al. 2024) trains early exit models with layer dropout to improve the performance of self-speculative decoding. BEEM (Bajpai and Hanawal 2025) treats multi-exit classifiers as experts and aggregates their confidence scores when neighboring experts show consistent predictions. COSEE (He et al. 2025) uses a calibrated weighting mechanism to ensure each classifier emphasizes samples likely to exit at the corresponding layer.

Despite the progress in early exit methodologies, existing approaches face fundamental limitations in trading off between accuracy and efficiency. Most conventional training schemes suffer from gradient conflicts when optimizing multiple exits simultaneously, where improvements in early exit performance come at the expense of degraded late exit accuracy. EnViT addresses these critical limitations through a novel training method to effectively enhance early exits and preserve the performance of late exits.

3 Method

This section presents the design of EnViT, including the preliminaries, the exit-aware structured dropout mechanism, and the novel training approach based on virtual sub-models.

Preliminaries

Early-Exit ViT Architecture. We adopt a Vision Transformer (ViT) backbone with L sequential layers. Let $\mathbf{h}_\ell$ denote the feature representations at layer ℓ. The forward propagation follows:

$$\mathbf{h}_\ell = \begin{cases} \mathcal{E}(\mathbf{x}), & \text{if } \ell = 0 \\ \mathcal{F}^\ell(\mathbf{h}_{\ell-1}), & \text{if } 1 \le \ell \le L \end{cases} \tag{1}$$

where $\mathcal{E}(\cdot)$ is the patch embedding function and $\mathcal{F}^\ell(\cdot)$ represents the ℓ-th Transformer layer.

We incorporate early exits by adding M classifiers at layers $\mathcal{S} = \{s_1, s_2, \ldots, s_M\}$ with $s_1 < s_2 < \ldots < s_M = L$. The i-th exit produces predictions as:

$$\hat{\mathbf{y}}_i = \mathcal{C}_i(\mathbf{h}_{s_i}) \tag{2}$$

where $\mathcal{C}_i(\cdot)$ is the classifier for the i-th exit.

Conventional Training Scheme. Traditional methods optimize the weighted sum of losses from all exits:

$$\mathcal{L} = \sum_{i=1}^{M} w_i \mathcal{L}_{\text{CE}}(\hat{\mathbf{y}}_i, \mathbf{y}) \tag{3}$$

This approach suffers from gradient conflicts due to overlapping optimization paths and provides limited supervision for early exits.

Exit-Aware Structured Dropout

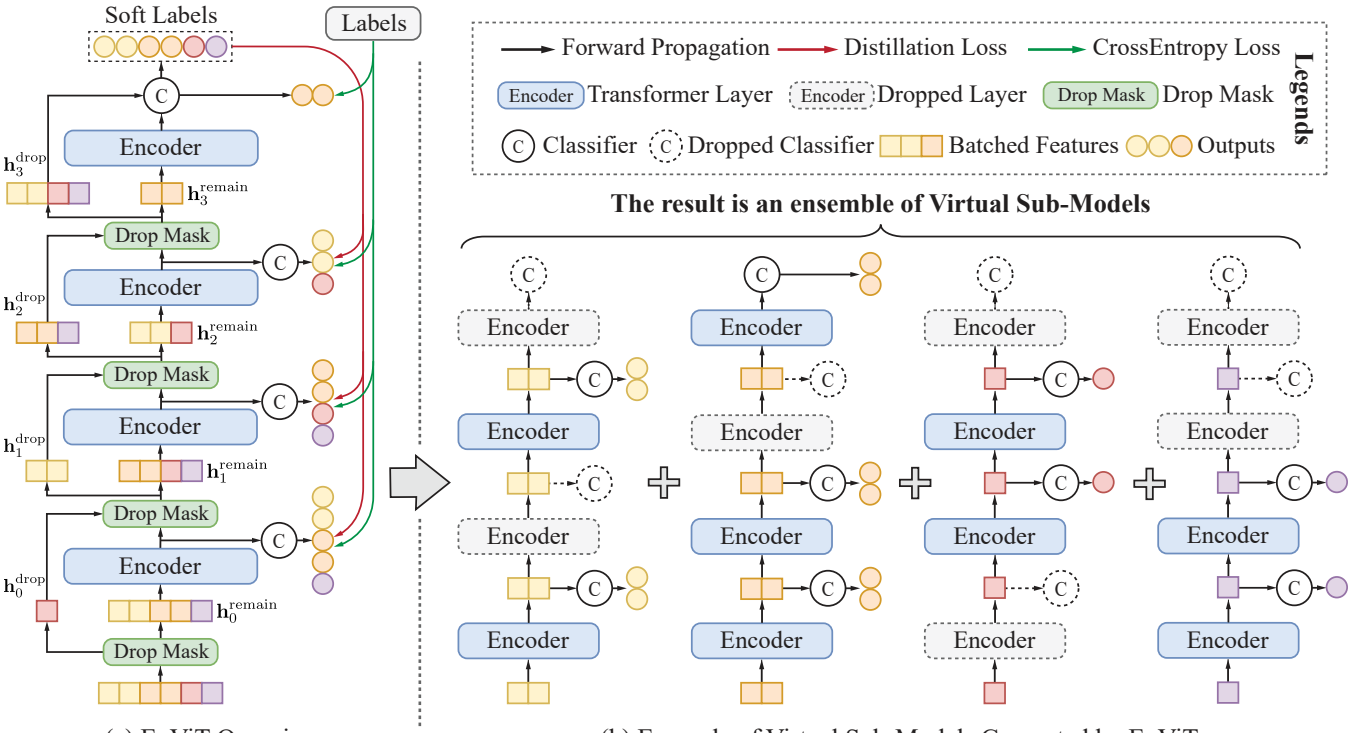
We first briefly introduce our new exit-aware structured dropout method. Then, we demonstrate how it generates virtual sub-models for a given batch and illustrate how it mitigates gradient conflicts through this perspective.

As discussed in §1, incorporating multiple exits into a model often leads to significant accuracy degradation. Existing training approaches for EE models simply sum the losses from all samples within a batch across all exits (Teerapittayanon, McDanel, and Kung 2016; Xu et al. 2023; He et al. 2025). When gradients from different exits propagate backward to the same layer, they exhibit substantial overlap, which leads to gradient conflicts (Gong et al. 2024). Introducing knowledge distillation adds additional gradient propagation paths, further exacerbating the problem (Zhang et al. 2019).

Structured Dropout within Batch. Dropout (Srivastava et al. 2014) is a widely-used regularization technique that randomly sets a fraction of neurons to zero during model training. It forces the model to not rely on specific neuron combinations. Inspired by this concept, instead of dropping individual neurons or features, EnViT performs structured dropout on each transformer layer as a whole with different probabilities based on exit positions. At every layer of the model, EnViT independently performs structured dropout for each sample within the same batch.

For each transformer layer ℓ, EnViT generates a sample-wise binary dropout mask $\mathbf{m}_\ell \in \{0, 1\}^B$ to control layer-wise execution during training:

$$m_\ell^{(i)} \sim \text{Bernoulli}(p_\ell), \quad i \in \{1, 2, \ldots, B\} \tag{4}$$



(a) EnViT Overview (b) Example of Virtual Sub-Models Generated by EnViT

Figure 2: Overview of the EnViT training framework. (a) EnViT applies exit-aware structured dropout during training. Different samples (represented by colored numbers) follow different computational paths based on dropout masks. (b) EnViT can be viewed as effectively training an ensemble of virtual sub-models, where each sub-model processes a subset of samples. This approach mitigates gradient conflicts while improving early exit performance.

where $m_\ell^{(i)}$ represents the dropout decision for sample i at layer ℓ, p_ℓ is the structured dropout probability assigned to layer ℓ, and B denotes the batch size. This mask determines whether layer ℓ will be skipped ($m_\ell^{(i)} = 1$) or executed ($m_\ell^{(i)} = 0$) for each sample in the batch. It enables sample-specific structured dropout within the same batch during training. The input features $h_{\ell-1}$ for layer ℓ is processed with the dropout mask:

$$\mathbf{h}_{\ell-1}^{\text{drop}} = \{\mathbf{h}_{\ell-1}^{(i)} \mid m_\ell^{(i)} = 1, i = 1, 2, \ldots, B\} \quad (5)$$

$$\mathbf{h}_{\ell-1}^{\text{remain}} = \{\mathbf{h}_{\ell-1}^{(i)} \mid m_\ell^{(i)} = 0, i = 1, 2, \ldots, B\} \quad (6)$$

where $h_{\ell-1}^{\text{remain}}$ are the features from samples that will be processed by layer ℓ, and $h_{\ell-1}^{\text{drop}}$ are the features from samples that will skip layer ℓ. The remaining features are then processed by the transformer layer:

$$\mathbf{h}_\ell^{\text{remain}} = \mathcal{F}^\ell(\mathbf{h}_{\ell-1}^{\text{remain}}) \quad (7)$$

If layer ℓ has an exit (i.e., $\ell \in \mathcal{S}$), the exit logits are computed from the remaining features:

$$\hat{\mathbf{z}}_\ell^{\text{remain}} = \mathcal{C}_\ell(\mathbf{h}_\ell^{\text{remain}}) \quad (8)$$

After processing, the remaining features are merged with the dropped features to produce the input for layer $\ell + 1$:

$$\mathbf{h}_\ell = \text{merge}(\mathbf{h}_\ell^{\text{remain}}, \mathbf{h}_{\ell-1}^{\text{drop}}) \quad (9)$$

The order of the samples in the batch will not change to facilitate the subsequent calculation of label loss and knowledge distillation loss.

Exit-Aware Dropout Probability. EnViT calculates the structured dropout probabilities by considering the positions of exit points. The layers between two consecutive exits share the same structured dropout probability. To encourage more samples to exit from early exits and improve the accuracy of early exits, EnViT assigns smaller probabilities to earlier layers and larger probabilities to later layers:

$$p_\ell = \begin{cases} 0, & \text{if } 0 < \ell \leq s_1 \\ \frac{j-1}{M-1} p_M, & \text{if } s_{j-1} < \ell \leq s_j \end{cases} \quad (10)$$

where p_M is the maximum structured dropout probability applied to the last layer, $j \in \{2, 3, \ldots, M\}$ denotes the exit index, and M is the total number of exits.

Understanding with Virtual Sub-Models. As illustrated in Figure 2, this sample-level structured dropout training approach dynamically creates different sub-models for multiple sub-batches within a single batch during training. Samples in different sub-batches pass through distinct virtual sub-models and exit from different exit points. This effectively reduces sample gradient overlap between different exits within a batch during backpropagation, thereby mitigating gradient conflicts. Furthermore, by assigning low prob-

abilities to early layers and high probabilities to late layers, more samples are encouraged to exit at early exits.

Training on Virtual Sub-Models

Our training strategy operates on virtual sub-models dynamically generated through exit-aware structured dropout. Unlike conventional multi-exit training that uses identical paths for all samples, the training result produced by EnViT is an ensemble of all virtual sub-models.

Enabling Self-Distillation. As discussed in §1, incorporating knowledge distillation directly during early exit model training compromises accuracy at late exits while providing only marginal accuracy improvements for early exits. It fails to achieve a proper tradeoff between inference accuracy and inference efficiency. This problem stems from knowledge distillation introducing additional gradient propagation paths, which exacerbates gradient conflicts.

Instead of directly distilling the final exit to previous exits, EnViT enables self-distillation by treating the complete model as teacher and virtual sub-models generated by structured dropout as students. It uses the full features h_L from the final layer and the corresponding classifier to obtain soft labels for all samples in a batch:

$$\hat{\mathbf{y}}_L = \text{softmax}(\mathcal{C}_L(\mathbf{h}_L)/\tau) \tag{11}$$

where τ is the temperature parameter for knowledge distillation. Before distilling these soft labels to each exit, the soft labels are processed in a similar way according to the mask of that exit:

$$\hat{\mathbf{y}}_{s_j}^{\text{soft}} = \{\hat{\mathbf{y}}_L^{(i)} \mid m_{s_j}^{(i)} = 0, i = 1, 2, \ldots, B\} \tag{12}$$

where $\hat{\mathbf{y}}_{s_j}^{\text{soft}}$ represents the filtered soft labels for exit j, and $j \in \{1, 2, \ldots, M-1\}$. The soft labels are then used to compute the distillation loss with the samples at exit s_j:

$$\mathcal{L}_{s_j}^{\text{distill}} = \tau^2 \cdot \text{KL}(\text{softmax}(\hat{\mathbf{z}}_{s_j}^{\text{remain}}/\tau), \hat{\mathbf{y}}_{s_j}^{\text{soft}}) \tag{13}$$

where $\hat{\mathbf{z}}_{s_j}^{\text{remain}}$ denotes the logits produced by the j-th exit from the remaining samples, $\text{KL}(\cdot, \cdot)$ denotes the Kullback-Leibler divergence. Through this distillation loss, EnViTs transfers the teacher's knowledge to all virtual sub-models that contain exit e_j.

Loss from Hard Labels. Similarly, to compute the loss on ground truth labels for samples at each exit, EnViT filters the sample labels according to the mask at each exit:

$$\mathbf{y}_{s_j}^{\text{remain}} = \{\mathbf{y}^{(i)} \mid m_{s_j}^{(i)} = 0, i = 1, 2, \ldots, B\} \tag{14}$$

where $\mathbf{y}_{s_j}^{\text{remain}}$ contains the ground-truth labels corresponding to the samples exit at s_j. The cross-entropy loss for each exit is then computed as:

$$\mathcal{L}_{s_j}^{\text{hard}} = \text{CrossEntropy}(\hat{\mathbf{z}}_{s_j}^{\text{remain}}, \mathbf{y}_{s_j}^{\text{remain}}) \tag{15}$$

where $j \in \{1, 2, \ldots, M\}$. Similar to self-distillation, the knowledge from hard labels is also transferred to all virtual sub-models that contain the corresponding exit.

Overall Training Objectives For each exit, the loss comprises the cross-entropy loss from hard labels and the KL Divergence loss from soft labels of the last exit. EnViT combines these losses to form the total loss for each exit:

$$\mathcal{L}_{s_j} = (1-\alpha)\mathcal{L}_{s_j}^{\text{hard}} + \alpha\mathcal{L}_{s_j}^{\text{distill}} \tag{16}$$

where α is a hyper-parameter used to balance different loss components. Specifically, for the final exit s_M, only the cross-entropy loss from hard labels is applied.

The overall optimization target is the weighted average of the losses from all exits:

$$\mathcal{L} = \frac{1}{M}\sum_{j=1}^{M}\frac{\mathcal{L}_{s_j}}{1-p_{s_j}} \tag{17}$$

where p_{s_j} is the structured dropout probability at layer s_j. Factor $\frac{1}{1-p_{s_j}}$ serves to compensate for the reduced number of samples at each exit due to structured dropout.

Early Exit Condition

EnViT uses the confidence score as the exit condition. For each sample at exit j, EnViT computes the confidence score as the maximum probability of the predicted class:

$$c_j = \max(\text{softmax}(\hat{\mathbf{z}}_{s_j})) \tag{18}$$

A sample exits at layer s_j if its confidence score exceeds a threshold. Otherwise, the sample continues to the next exit point for further processing. The speed-accuracy trade-off can be controlled by adjusting the threshold parameters.

Theoretical Analysis

Theorem 3.1 (Gradient Scaling)**.** *Under the exit-aware structured dropout mechanism in EnViT, the expected gradient received by layer ℓ is scaled by a factor of $(1-p_\ell)$ compared to the original gradient without dropout, where p_ℓ is the structured dropout probability at layer ℓ.*

Theorem 3.2 (Regularization Effect)**.** *EnViT adds implicit layer-specific regularization to the network that guides the training process of early-exit models, encouraging layers to specialize for samples of varying complexity.*

These theoretical results explain why EnViT achieves superior performance. Early exits benefit from lower dropout rates and larger gradients, enabling them to learn better discriminative features for early classification. Late exits receive smaller gradients that prevent over-optimization while preserving their capability to handle complex samples. Meanwhile, the regularization effect encourages layer-wise specialization, leading to more effective utilization of the network's capacity and improved accuracy-efficiency tradeoffs. The proofs of these theorems are provided in Appendix A.1 and A.2.

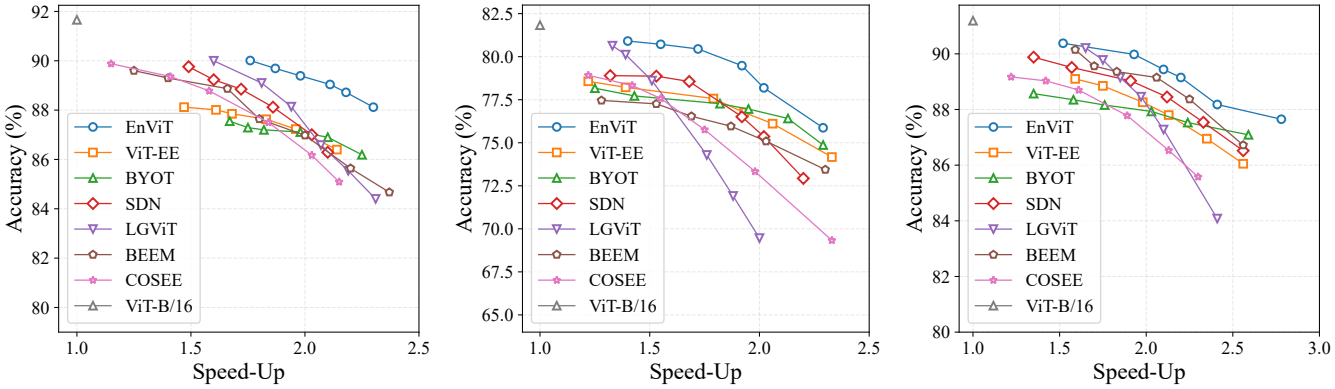
4 Experiments

Experiment Setup

Datasets. We evaluate EnViT on five commonly used public datasets: CIFAR-100 (Krizhevsky, Hinton et al. 2009), ImageNet-1k (Deng et al. 2009), Food-101 (Bossard, Guillaumin, and Van Gool 2014), Stanford Cars (Krause et al.

Methods	CIFAR-100		ImageNet-1k		Food-101		Stanford Cars		Flowers-102	
	Acc.	Speed-up	Acc.	Speed-up	Acc.	Speed-up	Acc.	Speed-up	Acc.	Speed-up
ViT-B/16	91.66%	1.00x	81.63%	1.00x	91.19%	1.00x	87.75%	1.00x	97.67%	1.00x
ViT-EE	87.63%	1.83x	77.57%	1.79x	88.85%	1.75x	83.41%	2.04x	94.26%	2.13x
SDN	88.12%	1.86x	78.56%	1.68x	88.45%	2.12x	85.45%	2.01x	95.07%	2.15x
BYOT	87.20%	1.82x	77.50%	1.63x	87.93%	2.03x	86.05%	2.01x	94.89%	2.00x
LGViT	88.14%	1.94x	78.61%	1.51x	88.46%	1.97x	79.11%	1.45x	93.14%	1.63x
BEEM	88.87%	1.66x	76.54%	1.69x	89.15%	2.06x	83.70%	1.70x	93.04%	2.14x
COSEE	87.50%	1.84x	76.67%	1.65x	87.78%	1.89x	80.94%	1.72x	91.96%	1.48x
Ours	**89.23%**	**2.05x**	**80.45%**	**1.72x**	**89.15%**	**2.20x**	**87.03%**	**2.05x**	**95.33%**	**2.23x**

Table 2: Performance comparison across different methods and datasets. "Acc." represents the top-1 classification accuracy. "Speed-up" represents the computational acceleration compared to the original model without early exit mechanisms.



(a) CIFAR-100 (b) ImageNet-1k (c) Food-101

Figure 3: Accuracy and efficiency trade-off comparison across datasets. EnViT significantly outperforms other methods.

2013) and Flowers-102 (Nilsback and Zisserman 2008). For datasets with labeled test sets, we conduct testing directly on the provided test set. Otherwise, we utilize the validation set instead. A summary of the datasets, including their classes, training, testing, and validation splits, is provided in Table 3.

Dataset	Classes	|Train |	|Test |	|Valid |
CIFAR-100	100	50k	10k	N/A
ImageNet-1k	1000	1.28M	100k	50k
Food-101	101	75.8k	N/A	25.3k
Stanford Cars	196	8.14k	8.04k	N/A
Flowers-102	102	1.02k	6.15k	1.02k

Table 3: Overview of the datasets used in the experiments.

Evaluation metrics. We compare the accuracy and speed-up under the trade-off between performance and efficiency for the entire model. For accuracy, we use Top-1 classification accuracy. For speed, to maintain consistency across different experiments and eliminate interference, following prior work (Xu et al. 2023; Bajpai and Hanawal 2025; He et al. 2025), we use the speed-up ratio. For an L-layer model, the speed-up ratio is computed as:

$$\text{Speed-up} = \frac{N \times C_L}{\sum_{i=1}^{L} n_i \times C_i} \tag{19}$$

where N is the total number of samples, n_i represents the number of samples that exit at the i-th layer, and C_i is the computational cost required to run up to the i-th layer. The actual inference acceleration is proportional to this speed-up ratio. We adopt Multiply-Accumulate Operations (MACs) as the computational cost metric.

Baselines. We compare EnViT against several baselines:

- **ViT-EE:** Directly adding classification heads to ViT and training with the sum of losses from all exits. (Bakhtiarnia, Zhang, and Iosifidis 2021)
- **SDN:** This method incorporates internal classifiers at intermediate layers and utilizes a weighted training strategy. (Kaya, Hong, and Dumitras 2019)
- **BYOT:** Adding multiple exits to ViT and training with self-distillation approach. (Zhang et al. 2019)
- **LGViT:** An early exit framework that incorporates local perception head and global aggregation head with a two-stage training scheme, adding a substantial number of parameters. (Xu et al. 2023)

- **BEEM:** A method that treats exit classifiers as experts and aggregates their scores when neighboring experts show consistent predictions. (Bajpai and Hanawal 2025)
- **COSEE:** A method that uses a calibrated weighting mechanism to enable each classifier to emphasize samples likely to exit at that classifier. (He et al. 2025)

Implementation. We implement EnViT with PyTorch and the Huggingface transformers library. For training, we use parameters pretrained on ImageNet and the AdamW optimizer. We place exits at every other layers and set the batch size to 128, learning rate to 5e-5, distillation loss weight α to 0.5, distillation temperature τ to 4 and probability p_M to 0.3. During inference, we achieve different accuracy and speed-up by adjusting the threshold. The experiments are conducted on a server equipped with two Intel Xeon Platinum 8352V CPUs (2.10 GHz), 256 GB RAM, and four NVIDIA A40 GPUs (48 GB each).

Performance Evaluation

We conduct comprehensive experiments on five widely-used public datasets to demonstrate the effectiveness of EnViT against six early-exit baselines.

Overall Performance Analysis. Table 2 presents the comprehensive comparison results across all five datasets. EnViT consistently outperforms existing approaches in terms of both classification accuracy and inference speed-up. On CIFAR-100, EnViT achieves a 89.23% accuracy with 2.05x speed-up, surpassing the state-of-the-art LGViT by 1.09% in accuracy and 0.11x in speed-up. Similar improvements are observed across other datasets: EnViT achieves the highest accuracy of 80.45% with substantial speed-up of 1.72x on ImageNet-1k, obtains comparable accuracy of 89.15% while delivering the highest speed-up of 2.20x on Food-101, reaches 87.03% accuracy with significant speed-up of 2.05x on Stanford Cars, and attains 95.33% accuracy with notable speed-up of 2.23x on Flowers-102.

Accuracy-Efficiency Trade-off Analysis. Figure 3 illustrates the accuracy vs. speed-up curves for three representative datasets CIFAR-100, ImageNet-1k and Food-101. These curves are generated by varying the exit threshold, allowing us to explore the full spectrum of accuracy-efficiency trade-offs. EnViT consistently dominates the trade-off, achieving higher accuracy at any given speed-up level compared to the baselines. Its superior performance stems from its exit-aware structured dropout-enabled self-distillation technique, which effectively enhances early exit capabilities while preserving the model's ability to handle complex samples through late exits.

Ablation Studies

We systematically analyze the contributions of structured dropout and self-distillation mechanisms by evaluating four different configurations: the baseline early-exit model, EnViT with only structured dropout (EnViT-SD), EnViT with only self-distillation (EnViT-KD), and the complete version of EnViT incorporating both components. Table 4 presents the results on CIFAR-100.

Config	Accuracy (%) of Exit					
	s_1	s_2	s_3	s_4	s_5	s_6
Baseline	60.48	75.61	83.13	87.21	88.85	89.40
EnViT-SD	60.51	76.23	83.57	87.19	89.12	89.88
EnViT-KD	61.37	78.05	83.46	86.51	87.68	87.73
EnViT	**61.94**	**79.02**	**84.67**	**87.72**	**88.85**	**89.94**

Table 4: Ablation study on CIFAR-100 with ViT-B/16 backbone. EnViT-SD denotes EnViT with structured dropout only, EnViT-KD denotes EnViT with self-distillation only, and EnViT represents the complete method with both components. All results show top-1 accuracy (%) at each exit.

Individual Component Analysis. To gain deeper insights into the contribution of different components to EnViT, we evaluate the accuracy of each of the six exits individually. The baseline model achieves an accuracy ranging from 60.48% at the first exit to 89.40% at the final exit,

When applying only structured dropout (i.e., EnViT-SD), we observe modest improvements across most exits with improvements of +0.03% at s_1, +0.62% at s_2, +0.44% at s_3, +0.27% at s_5 and +0.48% at s_6. In contrast, incorporating only self-distillation (EnViT-KD) yields more substantial improvements in early exits but reveals a critical limitation. While early exits show notable gains (+0.89% at s_1, +2.44% at s_2), the performance at late exits deteriorates significantly. Most concerning is the substantial degradation at s_4, s_5, and s_6, with accuracy drops of -0.70%, -1.17% and -1.67% respectively. This also validates our hypothesis that knowledge distillation exacerbates gradient conflicts and compromises late exit performance, limiting the model's ability to handle complex samples that require deeper processing.

Synergistic Effects. The complete EnViT, combining both structured dropout and self-distillation, achieves the best performance across all exits. Early exits benefit substantially from the enhanced supervision provided by self-distillation, achieving an improvement of +1.46% at s_1, +3.41% at s_2, +1.54% at s_3, and +0.51% at s_4 compared to the baseline. This shows that the structured dropout mechanism successfully mitigates the gradient conflicts, enabling the model to maintain high performance at late exits.

5 Conclusion

In this paper, we identified the key challenges in early-exit ViT, i.e., limited performance of early exits and accuracy degradation of late exits. To address these issues, we proposed EnViT, an exit-aware structured dropout-enabled self-distillation approach for training ViTs. By training on virtual sub-models generated by exit-aware structured dropout, EnViT enables effective knowledge transfer while mitigating gradient conflicts between different exits. Comprehensive experiments demonstrate that, compared with the state of the art, EnViT achieves accuracy improvements ranging from 0.36% to 7.92% while maintaining competitive speed-ups of 1.72x to 2.23x. It provides a practical solution for deploying ViTs on resource-constrained edge devices.

Acknowledgments

We sincerely thank the AC and reviewers for their constructive and valuable feedback. This research was supported by the National Key R&D Program of China under Grant No. 2023YFB4502400.

References

Bajpai, D. J.; and Hanawal, M. K. 2025. BEEM: Boosting Performance of Early Exit DNNs using Multi-Exit Classifiers as Experts. In *The Thirteenth International Conference on Learning Representations*.

Bakhtiarnia, A.; Zhang, Q.; and Iosifidis, A. 2021. Multi-Exit vision Transformer for Dynamic Inference. *arXiv preprint arXiv:2106.15183*.

Bossard, L.; Guillaumin, M.; and Van Gool, L. 2014. Food-101–Mining Discriminative Components with Random Forests. In *European Conference on Computer Vision*, 446–461.

Cao, Q.; Paranjape, B.; and Hajishirzi, H. 2023. PuMer: Pruning and Merging Tokens for Efficient Vision Language Models. In *61st Annual Meeting of the Association for Computational Linguistics*, 12890–12903.

Chefer, H.; Gur, S.; and Wolf, L. 2021. Transformer Interpretability beyond Attention Visualization. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 782–791.

Chen, M.; Peng, H.; Fu, J.; and Ling, H. 2021. Autoformer: Searching Transformers for Visual Recognition. In *IEEE/CVF International Conference on Computer Vision*, 12270–12280.

Deng, J.; Dong, W.; Socher, R.; Li, L.-J.; Li, K.; and Fei-Fei, L. 2009. ImageNet: A Large-scale Hierarchical Image Database. In *IEEE Conference on Computer Vision and Pattern Recognition*, 248–255.

Dosovitskiy, A.; Beyer, L.; Kolesnikov, A.; Weissenborn, D.; Zhai, X.; Unterthiner, T.; Dehghani, M.; Minderer, M.; Heigold, G.; Gelly, S.; Uszkoreit, J.; and Houlsby, N. 2021. An Image is Worth 16x16 Words: Transformers for Image Recognition at Scale. In *International Conference on Learning Representations*.

Elhoushi, M.; Shrivastava, A.; Liskovich, D.; Hosmer, B.; Wasti, B.; Lai, L.; Mahmoud, A.; Acun, B.; Agarwal, S.; Roman, A.; et al. 2024. LayerSkip: Enabling Early Exit Inference and Self-Speculative Decoding. *arXiv preprint arXiv:2404.16710*.

Gong, C.; Chen, Y.; Luo, Q.; Lu, Y.; Li, T.; Zhang, Y.; Sun, Y.; and Zhang, L. 2024. Deep Feature Surgery: Towards Accurate and Efficient Multi-exit Networks. In *European Conference on Computer Vision*, 435–451.

Graves, A. 2016. Adaptive Computation Time for Recurrent Neural Networks. *arXiv preprint arXiv:1603.08983*.

Han, S.; Mao, H.; and Dally, W. J. 2015. Deep Compression: Compressing Deep Neural Networks with Pruning, Trained Quantization and Huffman Coding. *arXiv preprint arXiv:1510.00149*.

He, J.; Zhang, Q.; Zhang, H.; Huang, X.; Naseem, U.; and Miao, D. 2025. Cosee: Consistency-Oriented Signal-Based Early Exiting via Calibrated Sample Weighting Mechanism. In *AAAI Conference on Artificial Intelligence*, 24023–24031.

Hinton, G.; Vinyals, O.; and Dean, J. 2015. Distilling the Knowledge in a Neural Network. *arXiv preprint arXiv:1503.02531*.

Huang, G.; Chen, D.; Li, T.; Wu, F.; van der Maaten, L.; and Weinberger, K. 2018. Multi-Scale Dense Networks for Resource Efficient Image Classification. In *International Conference on Learning Representations*.

Kaya, Y.; Hong, S.; and Dumitras, T. 2019. Shallow-Deep Networks: Understanding and Mitigating Network Overthinking. In *International Conference on Machine Learning*, 3301–3310.

Kong, Z.; Dong, P.; Ma, X.; Meng, X.; Niu, W.; Sun, M.; Shen, X.; Yuan, G.; Ren, B.; Tang, H.; et al. 2022. Spvit: Enabling Faster Vision Transformers via Latency-Aware Soft Token Pruning. In *European Conference on Computer Vision*, 620–640.

Krause, J.; Stark, M.; Deng, J.; and Fei-Fei, L. 2013. 3d Object Representations for Fine-Grained Categorization. In *IEEE International Conference on Computer Vision Workshops*, 554–561.

Krizhevsky, A.; Hinton, G.; et al. 2009. Learning Multiple Layers of Features from Tiny Images. *University of Toronto*.

Liu, W.; Zhou, P.; Wang, Z.; Zhao, Z.; Deng, H.; and Ju, Q. 2020. FastBERT: a Self-distilling BERT with Adaptive Inference Time. In *58th Annual Meeting of the Association for Computational Linguistics*, 6035–6044.

Liu, Z.; Wang, Y.; Han, K.; Zhang, W.; Ma, S.; and Gao, W. 2021. Post-Training Quantization for Vision Transformer. *Advances in Neural Information Processing Systems*, 34: 28092–28103.

Mehta, S.; and Rastegari, M. 2022. MobileViT: Lightweight, General-purpose, and Mobile-friendly Vision Transformer. In *International Conference on Learning Representations*.

Meng, L.; Li, H.; Chen, B.-C.; Lan, S.; Wu, Z.; Jiang, Y.-G.; and Lim, S.-N. 2022. Adavit: Adaptive Vision Transformers for Efficient Image Recognition. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 12309–12318.

Nilsback, M.-E.; and Zisserman, A. 2008. Automated Flower Classification over a Large Number of Classes. In *Sixth Indian Conference on Computer Vision, Graphics & Image Processing*, 722–729.

Rao, Y.; Zhao, W.; Liu, B.; Lu, J.; Zhou, J.; and Hsieh, C.-J. 2021. Dynamicvit: Efficient Vision Transformers with Dynamic Token Sparsification. *Advances in Neural Information Processing Systems*, 34: 13937–13949.

Sajjad, H.; Dalvi, F.; Durrani, N.; and Nakov, P. 2023. On the Effect of Dropping Layers of Pre-Trained Transformer Models. *Computer Speech & Language*, 77: 101429.

Srivastava, N.; Hinton, G.; Krizhevsky, A.; Sutskever, I.; and Salakhutdinov, R. 2014. Dropout: a Simple Way to Prevent

Neural Networks from Overfitting. *The journal of Machine Learning Research*, 15(1): 1929–1958.

Teerapittayanon, S.; McDanel, B.; and Kung, H.-T. 2016. Branchynet: Fast Inference via Early Exiting from Deep Neural Networks. In *International Conference on Pattern Recognition*, 2464–2469.

Vaswani, A.; Shazeer, N.; Parmar, N.; Uszkoreit, J.; Jones, L.; Gomez, A. N.; Kaiser, Ł.; and Polosukhin, I. 2017. Attention is All You Need. *Advances in neural information processing systems*, 30.

Wang, X.; Yu, F.; Dou, Z.-Y.; Darrell, T.; and Gonzalez, J. E. 2018. Skipnet: Learning Dynamic Routing in Convolutional Networks. In *European Conference on Computer Vision*, 409–424.

Wu, K.; Zhang, J.; Peng, H.; Liu, M.; Xiao, B.; Fu, J.; and Yuan, L. 2022. Tinyvit: Fast Pretraining Distillation for Small Vision Transformers. In *European Conference on Computer Vision*, 68–85.

Xu, D.; Zhang, H.; Yang, L.; Liu, R.; Huang, G.; Xu, M.; and Liu, X. 2025a. Fast On-Device LLM Inference with NPUs. In *30th ACM International Conference on Architectural Support for Programming Languages and Operating Systems*, 445–462.

Xu, G.; Hao, J.; Shen, L.; Hu, H.; Luo, Y.; Lin, H.; and Shen, J. 2023. LGViT: Dynamic Early Exiting for Accelerating Vision Transformer. In *31st ACM International Conference on Multimedia*, 9103–9114.

Xu, M.; Cai, D.; Yin, W.; Wang, S.; Jin, X.; and Liu, X. 2025b. Resource-Efficient Algorithms and Systems of Foundation Models: A Survey. *ACM Computing Surveys*, 57(5): 1–39.

Yin, H.; Vahdat, A.; Alvarez, J. M.; Mallya, A.; Kautz, J.; and Molchanov, P. 2022. A-vit: Adaptive Tokens for Efficient Vision Transformer. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 10809–10818.

Zhang, L.; Song, J.; Gao, A.; Chen, J.; Bao, C.; and Ma, K. 2019. Be Your Own Teacher: Improve the Performance of Convolutional Neural Networks via Self Distillation. *IEEE/CVF International Conference on Computer Vision*, 3712–3721.

Zheng, C.; Zhang, K.; Yang, Z.; Tan, W.; Xiao, J.; Ren, Y.; Pu, S.; et al. 2022. Savit: Structure-Aware Vision Transformer Pruning via Collaborative Optimization. *Advances in Neural Information Processing Systems*, 35: 9010–9023.

Zhou, W.; Xu, C.; Ge, T.; McAuley, J.; Xu, K.; and Wei, F. 2020. BERT Loses patience: Fast and Robust Inference with Early Exit. In *34th International Conference on Neural Information Processing Systems*, 18330–18341.

Zhu, W. 2021. LeeBERT: Learned Early Exit for BERT with Cross-Level Optimization. In *59th Annual Meeting of the Association for Computational Linguistics*, 2968–2980.

Zhu, W.; Wang, X.; Ni, Y.; and Xie, G. 2021. GAML-BERT: Improving BERT Early Exiting by Gradient Aligned Mutual Learning. In *Conference on Empirical Methods in Natural Language Processing*, 3033–3044.