Decision-centered teamwork in Dynamic Early Exiting Networks

Haoran Peng [1] **João Vitor de Carvalho Silva** [2] **Erickson R. Nascimento** [2] **Leandro Soriano Marcolino** [3 1]

Abstract

Dynamic early exiting is an effective method for improving the efficiency of deep neural networks. By adding classifiers into intermediate layers of deep neural networks, early exiting allows inference to conclude early for simpler samples. However, usually early exiting networks use the output of the final processed classifier, ignoring the information given by the previous classifiers. To address this, we explore decision-centered teamwork in dynamic networks. We propose a Bayesian updating method to effectively integrate information from earlier classifiers and introduce a conformal prediction-based approach to improve likelihood estimation. Additionally, we propose a logit voting method that, despite using less information, performs well in practical deep learning applications. It provides a practical alternative when accurate likelihoods for Bayesian updating are difficult to obtain. Through synthetic and real-world experiments, we demonstrate the effectiveness of these methods and the significant potential of the Bayesian updating approach.

1. Introduction

Although deep neural networks have achieved remarkable success across a wide range of tasks (Krizhevsky et al., 2012; He et al., 2016; Huang et al., 2019; Dosovitskiy, 2020), their computational demands present a significant challenge for deployment on resource-constrained devices. Various approaches have been explored to improve the inference efficiency of deep neural networks, such as network pruning (LeCun et al., 1989; He et al., 2019; Yang et al., 2021), weight quantization (Hubara et al., 2016; Han et al., 2015; Jung et al., 2019), and the development of lightweight architectures (Howard et al., 2017; Zhang et al., 2018; Sandler et al., 2018). Although these methods are effective, handling challenging data samples still requires large-scale deep neural networks. Consequently, dynamic neural networks (Han et al., 2021; Bolukbasi et al., 2017) have gained attention for the ability to perform data-dependent inference by dynamically adapting the models.

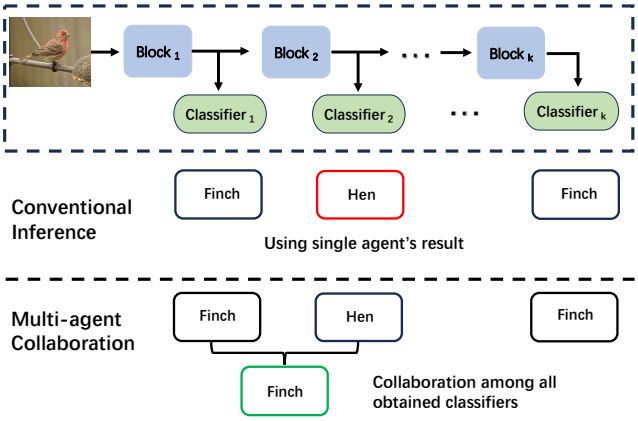


Figure 1. **Multi-agent collaboration-based inference strategy.** During inference, instead of relying solely on the deepest classifier, we utilize the collaborative output from all classifiers that have produced results so far.

Dynamic early exiting, a representative approach in dynamic neural networks, adds multiple classifiers to intermediate layers. It enables early termination of inference when predefined criteria, such as softmax confidence scores (Huang et al., 2017b) or a learned policy (Chen et al., 2020), are met. Current mainstream early exiting methods typically use the output of the deepest classifier at the exit point as the final result, ignoring the potential value of intermediate classifier outputs. However, these outputs, as suggested by Bayesian theory, can provide valuable contextual information to enhance model performance.

In the field of early exiting, collaboration among multiple classifiers has primarily focused on the training phase. For example, Han *et al.*(Han et al., 2022) and Sun *et al.*(Sun et al., 2022) proposed a meta-learning approach to address gradient conflicts between classifiers during training, while Wolczyk *et al.* (Wołczyk et al., 2021) introduced skip connections and knowledge distillation techniques to im-

[*]Equal contribution [1]Lancaster University, Lancaster, United Kingdom [2]Universidade Federal de Minas Gerais [3]VinUniversity, Hanoi, Vietnam. Correspondence to: Leandro Soriano Marcolino <leandro.sm@vinuni.edu.vn>.

Proceedings of the 42nd International Conference on Machine Learning, Vancouver, Canada. PMLR 267, 2025. Copyright 2025 by the author(s).

prove overall training performance. In the inference phase, Patience-based Early Exit (PBEE) (Zhou et al., 2020) proposed a novel exit mechanism where the model exits when consecutive classifiers produce the same output. Although this approach improves accuracy, it often fails to achieve a satisfactory trade-off between efficiency and accuracy.

In this paper, we explore multi-agent collaboration within dynamic early exiting networks. In particular, in decision-centered teamwork, agents solve challenges by taking joint decisions as a team (Marcolino et al., 2013; Marcolino, 2016; Marcolino et al., 2014). However, in deep learning, the application of ensemble methods faces one issue: training multiple agents (classifiers) often requires significant computational resources. Early exiting architectures, which have become increasingly important in the era of large models, inherently contain multiple classifiers, effectively avoiding the additional computational costs associated with ensemble methods. This makes early exiting a promising framework for exploring decision-centred teamwork in large-scale deep learning area.

Despite this potential, applying decision-centred teamwork in deep learning introduces new challenges. Large-scale deep learning tasks are often complex, such as classification problems with hundreds or even thousands of categories. To explore decision-centred teamwork in dynamic early exiting networks, we propose a Bayesian update-based collaboration method. Specifically, we obtain the likelihood of each classifier's output for different labels by counting and analyzing results from a validation set, which are then used for Bayesian updating. However, through synthetic experiments, we find that accurately estimating likelihoods when the validation set is not large is challenging, significantly limiting the performance of Bayesian updating in some scenarios.

To address this limitation, we propose a conformal prediction-based approach to improve the accuracy of likelihood estimation. I.e., we leverage the effectiveness of conformal prediction in large-scale deep learning tasks and use its more accurate confidence levels to refine the confidence estimates within the likelihood calculation.

Additionally, we propose a logit voting method as an alternative. While this approach utilizes less information and may have lower theoretical potential compared to Bayesian updating, it often performs well in practical deep learning scenarios, making it a viable fallback when accurate likelihoods for Bayesian updating are difficult to obtain.

Hence, in this paper, we make the following contributions:

1. **Decision-centered teamwork in dynamic neural networks**: We explore decision-centered teamwork in large-scale dynamic networks, a domain with significant challenges and opportunities.
2. **Bayesian updating method:** We propose a Bayesian updating-based collaboration method to enhance the interactions between agents in dynamic networks.
3. **Conformal prediction for likelihood improvement:** To address the inaccuracy of likelihood estimation in Bayesian updating, we introduce a conformal prediction-based approach that significantly improves the reliability of the method.
4. **Extensive experimental validation:** We conduct extensive experiments, including synthetic analyses and real-world computer vision applications, to demonstrate the effectiveness and strong potential of the Bayesian updating method.
5. **Voting-based alternative:** We study a voting-based method as a simpler alternative. Despite its simplicity, it performs well in practice and serves as a viable option when Bayesian updating is constrained by the size of the validation set.

Therefore, our work introduces a new challenge for the multi-agent systems community, with significant potential for broader applications in large-scale deep learning.

2. Related Works

2.1. Dynamic early exiting

Early exiting is a dynamic neural network architecture that allows simple inputs to exit early through intermediate classifiers. Some studies focus on the architectural design of early exiting models. BranchyNet (Teerapittayanon et al., 2016) places classifiers at multiple depths to enable early decisions and reduce computation. MSDNet (Huang et al., 2017b) provides classifiers with features from different levels to handle varying input complexities. RANet (Yang et al., 2020) uses conditional activation to process high-resolution data only when necessary, balancing accuracy and efficiency.

Additionally, ZTW (Wołczyk et al., 2021) emphasizes feature reuse, where features extracted by shallow classifiers are reutilized by deeper ones, effectively reducing errors not addressed by earlier classifiers. Beyond architecture, several studies have investigated exit strategies. For example, reinforcement learning has been utilized to learn effective exit policies (Huang et al., 2017a), while learned policy networks have been developed to optimize the exit decision (Chen et al., 2020).

PABEE (Zhou et al., 2020) introduces a simple yet effective early exiting mechanism based on output stability. It defines a patience score that monitors the consistency of

predictions across successive classifiers. When the same output is observed over a predefined number of consecutive layers, the network triggers an early exit. This strategy implicitly aggregates information from multiple classifiers and can improve prediction accuracy. However, the exit criterion is often overly strict, requiring multiple identical outputs before termination. As a result, it tends to delay exiting, leading to a suboptimal trade-off between accuracy and efficiency.

In contrast, our method directly leverages the predictions of earlier classifiers to enhance the performance of the current classifier, without modifying the existing exit mechanism. As a result, it achieves a more favorable trade-off between accuracy and efficiency. Moreover, our aggregation strategy goes beyond simple majority voting schemes by incorporating richer information than just the predicted labels.

2.2. Conformal Prediction

Conformal prediction (CP) provides a practical framework for uncertainty quantification by constructing prediction sets based on model confidence. In classification, CP uses a calibration set to compute a softmax-based threshold for a desired coverage level. At test time, it forms a prediction set by including labels whose cumulative softmax scores exceed this threshold, ensuring the true label is likely to be contained. Originating from the work of Vovk et al. (Vovk et al., 2005), CP operates under minimal assumptions—requiring only data exchangeability—and has been widely applied in classification, regression, and other machine learning tasks.

In classification, CP produces adaptive prediction sets that reflect the model's confidence, typically based on softmax-derived nonconformity scores (Angelopoulos & Bates, 2021; Shafer & Vovk, 2008). As a simple post hoc method, CP is particularly appealing in the deep learning era: it requires no additional training and utilizes only the outputs of a pretrained model. This makes it an efficient and scalable approach to uncertainty estimation in modern deep neural networks.

In our setting, likelihood prediction requires more fine-grained information than what is typically provided by standard conformal sets. As more conditioning factors are introduced, the amount of calibration data available for each condition decreases, which can significantly degrade the reliability of conformal prediction (Ding et al., 2023). Prior work by Angelopoulos et al. has shown that even when the conformal set contains only a single label, the method can still perform well (Angelopoulos et al.). Building on this insight, we adapt conformal prediction to our task by using only high-confidence, single-label conformal sets to enhance likelihood estimation.

2.3. Multi-agent collaboration

Multi-agent collaboration has been extensively studied in the context of reinforcement learning (RL) and other distributed systems. At its core, multi-agent collaboration involves multiple agents working together to achieve a shared goal, often requiring effective communication, coordination, and learning of joint policies.

In particular, in decision-centered teamwork, agents aggregate their opinions to take a joint team decision. Marcolino et al. (Marcolino et al., 2013; 2014) have shown that when there is sufficient diversity among agents, a voting-based approach can enable a group of relatively weaker agents to outperform a single, stronger agent. This method leverages the diversity within the group, allowing the collective decision-making process to mitigate individual weaknesses and harness the strengths of each agent, resulting in more robust and effective outcomes.

Regarding the Bayesian approach, a study has employed Bayesian updating in multi-agent collaboration to detect disruptive agents (do Carmo Alves et al., 2024). By iteratively updating beliefs based on observed behaviours, the system can identify and isolate potentially harmful individuals, enhancing overall group stability and performance. However, that work was aimed at a different coordination task, and did not tackle the decision-centered teamwork challenge as we do in this work. Additionally, they could estimate likelihoods from the outcomes of a Monte Carlo Tree Search process, which is not applicable in deep learning tasks.

3. Background

We start by introducing dynamic early exiting networks, and presenting the motivation for our work.

3.1. Dynamic Early Exiting Networks

Unlike static neural networks, dynamic early exiting networks add classifiers to intermediate layers of the original architecture. For each classifier C_c:

$$p_c = C_c\left(\mathbf{x}, \theta_c\right), \tag{1}$$

where $\mathbf{x}$ represents the input data sample, θ_c indicates the parameters of the c-th backbone and classifier. The prediction of the c-th classifier for the data sample $\mathbf{x}$ is denoted as p_c, representing the probability distribution over all classes. Each classifier consists of a fully connected layer followed by a softmax operation, which generates this probability distribution.

During the inference phase, each classifier produces an output along with a confidence score, typically represented by the maximum softmax value. If this confidence score exceeds a predefined threshold, the model exits early; other-

wise, it continues to process deeper layers.

3.2. Multi-agent collaboration

In dynamic multi-exit networks, classifiers are placed at different depths to handle inputs of varying difficulty—easy samples exit early, while harder ones proceed to deeper layers. Although these classifiers share a common backbone, each one makes decisions at different stages and with different levels of abstraction. This motivates a multi-agent perspective, where each classifier acts as an agent making independent decisions based on its local view.

However, unlike traditional ensembles that rely on fully independent models, our work investigates aggregation in a shared-backbone setting, which is specific to dynamic neural networks. That being said, parameter sharing does not negate the need for coordination. Prior work on decision-centered teamwork (Marcolino et al., 2013) shows that combining individual decisions can outperform isolated reasoning. Building on this insight, we introduce a framework that integrates multi-agent collaboration into dynamic networks, bridging two active research areas in a unified approach.

3.3. Motivations

In dynamic early exiting networks, the mainstream approach during inference is still to select the output of a single classifier p_c as the final result. When obtaining the output from the c-th classifier, the results from the 1st to the $(c-1)$th classifiers have already been generated but are discarded. Generally, deeper networks tend to offer better accuracy, however, studies have shown that dynamic early exiting networks may suffer from overthinking, where a shallow classifier produces the correct result while deeper classifiers make errors (Kaya et al., 2019). In the field of multi-agent collaboration, there are works suggesting that a team of diverse (and weak) agents can, through collaboration, outperform a single, stronger agent (Marcolino et al., 2013).

Bayesian updating is a promising approach for refining the probabilities of different options based on a series of observations. In image recognition tasks, analyzing the confusion matrix reveals that classifier errors often concentrate on certain classes. For instance, a model may frequently confuse cats with tigers but rarely with airplanes. In such cases, leveraging predictions from earlier shallow classifiers when the model produces easily confusable outputs can lead to more accurate results.

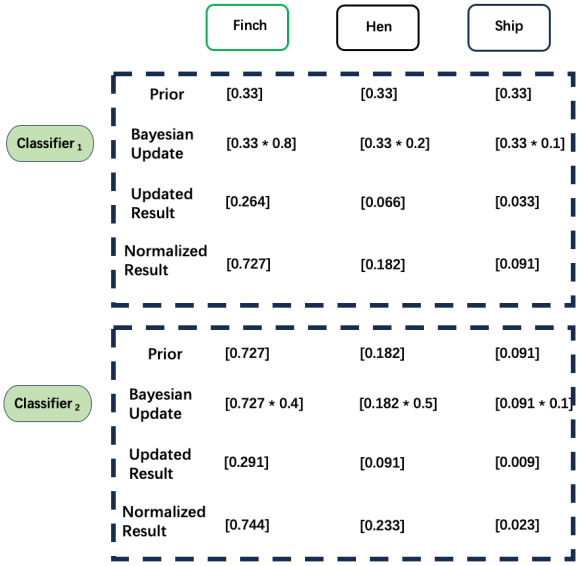


Figure 2. **An illustration of Bayesian Updating.**

4. Methodology

4.1. Bayesian updating based multi-classifiers collaboration

We propose a Bayesian updating method to facilitate collaboration between classifiers in dynamic early exiting networks. This method integrates outputs from all previous classifiers to calculate the probability of each class.

The Bayesian equation is reformulated as follows to compute the probability of label l_i given the output θ_c of classifier c:

$$P(l_i|\theta_c) = \frac{P(\theta_c|l_i) * P(l_i|\theta_{c-1})}{P(\theta_c)} \tag{2}$$

$P(l_i|\theta_c)$ represents the posterior distribution obtained in the current round of Bayesian updating, while $P(l_i|\theta_{c-1})$ is the posterior distribution from the previous round, which serves as the prior distribution for the current update. In the first round of Bayesian updating, the prior distribution is initialized as a uniform distribution across all labels. $P(\theta_c|l_i)$ represents the likelihood of the current classifier, which can be estimated based on the model's performance on the validation set. For the class probabilities $P(l_i|\theta_c)$ that we ultimately need, $P(\theta_c)$ is identical across all classes l_i, and thus can be normalized out. This process is applied recursively across classifiers in the dynamic neural network, updating the posterior distribution.

We use a simple 3-class classification example to illustrate our Bayesian updating method, as shown in Figure 2. The

prior distribution is initialized as uniform at the first classifier. The likelihood is derived from the output of the current classifier. The posterior distribution is then calculated by multiplying the likelihood with the prior and normalizing it to obtain a valid probability distribution. In subsequent rounds, the posterior from the previous round is used as the prior, and the process is repeated.

The likelihoods in our Bayesian updating method incorporate confidence levels and common misclassification patterns. For example, in Figure 2, Classifier 2 makes a common error, confusing 'hen' with 'finch', while being unlikely to confuse them with 'ship'. The likelihood from Classifier 2 assigns higher probability to 'hen' while also reflecting this common confusion. By leveraging prior information from earlier classifiers, which were more confident in 'finch', the error can be corrected.

4.2. Likelihood estimation

Here, we start with the simplest representation, where θ_c corresponds to the label with the highest confidence (e.g., the argmax of a softmax output). Later, we will introduce a more advanced version that incorporates softmax information for improved performance.

The likelihood $P(\theta_c \mid l_i)$ is estimated by deriving it from the confusion matrix of the corresponding classifier on the validation set, which shows how often each output is produced for each ground truth label. Specifically, for each classifier c, the confusion matrix records the frequency with which it outputs a specific label when the true label is l^*. Aggregating these frequencies across the validation set provides an approximation of $P(\theta_c \mid l_i)$ for all possible outputs and labels l_i.

Specifically, we use the validation set to construct a confusion matrix that records the count of each combination of output θ_c and its corresponding ground truth label l_i. This provides an estimate of $P(\theta_c \mid l_i)$, the probability of producing θ_c given that the ground truth is l_i. In our Bayesian updating process, θ_c represents the observed output of the current classifier. Using the information from the confusion matrix, we estimate $P(\theta_c \mid l_i)$ for different ground truth labels l_i.

4.3. Challenges of Bayesian Updating in Large-Scale Deep Learning

In the era of large models, deep learning is often used to tackle complex tasks and large datasets. This complexity makes it challenging to accurately estimate the likelihood in Bayesian updating. For instance, in classification tasks with hundreds or even thousands of classes, computing the likelihood from a validation set requires a significant amount of data to maintain accuracy. Additionally, discrepancies between the distribution of the validation set and the test set in real-world applications further impact the reliability of the likelihood estimates. For detailed analysis, please refer to our synthetic experiments.

We also explored the potential of Bayesian updating under ideal conditions by calculating likelihoods using the test set to construct the confusion matrix. This serves only as the upper bound performance under oracle likelihood conditions. Notably, this setup does not involve using the test set for training and therefore does not affect the model's performance. It only influences the confidence levels of the predictions. To ensure fairness, we retained the initial values for each class to prevent the model from exploiting exclusions to achieve better results. Our experiments demonstrate that Bayesian updating has significant potential, with the accuracy of likelihood estimation being the primary limiting factor. Confidence estimation in large-scale deep learning remains a challenging and open problem.

4.4. Enhanced Likelihood Estimation Through Conformal Prediction

Estimating likelihoods in Bayesian updating is difficult, especially in large-scale deep learning, which requires better methods for confidence estimation. Conformal prediction is a promising approach for confidence estimation, showing strong performance in large-scale deep learning scenarios (Angelopoulos & Bates, 2021). Moreover, it incorporates information from softmax outputs to provide a more precise definition of outputs in our Bayesian method, enhancing the accuracy of likelihood estimation.

However, conformal prediction does not directly provide a confidence score. By analyzing the softmax scores on a validation set, it generates conformal sets that capture the uncertainty of predictions (Angelopoulos et al.). Specifically, the method accumulates softmax probabilities until a predefined threshold is reached, at which point the corresponding labels are included in the conformal set. These sets ensure that the true label is contained with a predefined confidence level, making the predictions more reliable and interpretable.

We observed that conformal prediction maintains high confidence accuracy even for single outputs (i.e., when the conformal set is limited to one). Leveraging this property, we further refined our likelihood estimation method. Specifically, we categorized outputs into three groups based on predefined confidence levels (e.g., above 90%, 90%-80%, and below 80%) using their associated softmax values. That is, for each label l_i, a classifier may output $\theta_c = \{l_i, > 90\%\}$, $\theta_c = \{l_i, 80\% - 90\%\}$, or $\theta_c = \{l_i, < 80\%\}$. We then adjusted our likelihood computation by performing counts separately within each group.

By incorporating conformal prediction, we include softmax information in the output and achieve more accurate confidence estimates.

4.5. Logit Voting

List et al. (List & Goodin, 2001) extended the Condorcet Jury Theorem by showing that if each agent is more likely to give the correct answer than any particular incorrect one, then as the number of agents approaches infinity, the probability that majority voting yields the correct outcome converges to 1. This condition on individual accuracy is easily satisfied in large-scale deep learning settings. As a result, collaborative voting can still provide benefits, even when all participating agents are relatively weak.

Diversity has been widely recognized as a key factor for the effectiveness of voting-based methods. However, in the context of dynamic early exiting, diversity arises in a fundamentally different way. Unlike traditional ensembles composed of independent and uniformly strong models, dynamic networks generate a sequence of intermediate predictions with varying capacity. In this setting, collaboration is inherently constrained: each exit can only aggregate information from earlier, already computed predictions, which are typically less accurate but provide complementary views.

Definition 4.1 (Diversity in Dynamic Neural Networks)**.** We define *diversity* in a dynamic neural network as the occurrence of disagreement between classifiers at different exits, where an earlier (shallower) classifier produces the correct prediction while the current (deeper) classifier yields an incorrect one.

In dynamic neural networks, agents (i.e., exit classifiers) differ in their parameter capacity, leading to varying levels of performance. Deeper classifiers are typically more accurate due to their access to more expressive features. As a result, conventional measures of diversity—often designed for ensembles of uniformly strong models—are not directly applicable in this asymmetric setting. A dynamic neural network can benefit from multi-agent collaboration only when the form of diversity we define is present.

We also study the performance of a simple voting mechanism in Dynamic Neural Networks. In the voting mechanism, we employ a ranked-voting based approach, where the full logits output of each classifier is considered. Therefore, we calculate $O(\mathbf{x}) = \sum_c w_c * \theta_c(\mathbf{x})$, where here $\theta_c(\mathbf{x})$ represents the full logits output vector of a classifier c, given an item $\mathbf{x}$, and w_c is the weight assigned to the classifier c. The system then outputs the label l with the highest value in the vector $O(\mathbf{x})$.

Inspired by conformal prediction, we observe that deep neural networks inherently encode confidence information, often reflected in the magnitude of softmax outputs. Higher softmax scores are generally associated with higher confidence. However, using raw softmax values as confidence estimates can lead to overconfidence issues. We note that softmax is not directly optimized during training; instead, the loss is computed over the log-softmax (i.e., logits combined with the log function). As a result, the loss becomes highly sensitive when softmax values approach 1—small changes in softmax can lead to large changes in loss. Based on this observation, we propose to use logits directly for voting. Empirically, we find that logit-based voting significantly outperforms softmax-based voting.

Note that the full logits output represents how confident a classifier is at each possible output label. Therefore, labels which are assigned higher confidence will receive higher weight by the voting mechanism. This weight is then further multiplied by how much confidence we have on the classifier *per se* (w_c). We can employ a standard voting mechanism, where each classifier has the same weight ($w_c = 1$), or the performance of each classifier can be estimated in a validation set, in order to estimate the weights w_c. We find that in practical applications, assigning weights based on classifier performance often yields better results. This is particularly effective when certain classifiers perform significantly worse than others, as demonstrated in our experiments on the ImageNet dataset.

Contrary to usual ensemble-based approaches, we do not always consider the full set of available classifiers. As mentioned before, a dynamic neural network may decide to stop processing an item further, based on some policy. We then aggregate the outputs from the processed layers (classifiers), as the results from the unprocessed layers are not yet available.

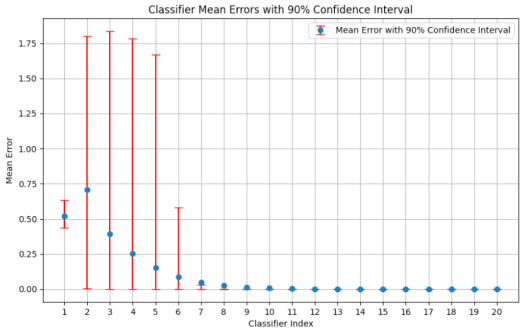
5. Experiments

In this section, we conduct extensive experiments on the CIFAR (Krizhevsky et al., 2009) and ImageNet (Deng et al., 2009) dataset, which is a representative dataset in computer vision, and pioneered the large-scale application of deep learning in image recognition. We demonstrate the effectiveness of our method on two representative dynamic early exiting network architectures, MSDNet (Huang et al., 2017b) and RANet (Yang et al., 2020). We also present detailed synthetic experiments to further analyze the factors influencing the Bayesian updating method. Additionally, we provide oracle results, assuming ideal likelihoods, to demonstrate the potential of the Bayesian updating approach.

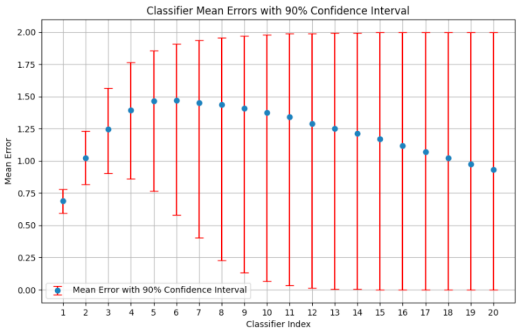
Datasets: Both CIFAR-100 and CIFAR-10 dataset (Krizhevsky et al., 2009) consists of 50,000 training images and 10,000 test images, each with a resolution of 32×32 pixels. CIFAR-100 includes 100 distinct classes for classification and CIFAR-10 have 10 classes. ImageNet (Deng

Table 1. **Anytime prediction results** of a 7-exit MSDNet on CIFAR-100

Method	1	2	3	4	5	6	7
Params($\times 10^6$)	0.30	0.65	1.11	1.73	2.38	3.05	4.00
FLOPs($\times 10^6$)	6.86	14.35	27.29	48.45	76.43	108.90	137.30
MSDNet	64.68	67.00	69.69	71.13	71.13	72.20	73.06
Bayes	64.68	66.32	70.45	72.57	73.78	74.47	75.00
Bayes + CP	64.68	67.80	70.85	73.04	73.96	74.61	75.02
Bayes + Oracle	64.68	70.34	73.36	75.25	76.14	77.10	77.66
Bayes + CP + Oracle	64.68	71.44	74.75	76.63	77.68	78.48	78.98
Logit Voting	64.68	69.84	72.53	73.91	74.64	75.28	75.60
FLOPs W. Voting	64.68	69.56	72.55	73.99	74.54	75.14	75.49
Acc W. Voting	64.68	69.86	72.64	73.87	74.74	75.29	75.59



((a)) **Simulation Analysis of CIFAR-100 experiment setting**



((b)) **Simulation Analysis of ImageNet experiment setting**

Figure 3. **Comparison of Synthetic Experiments for task scale of CIFAR-100 and ImageNet**

et al., 2009), on the other hand, comprises 1.2 million training images and 50,000 validation images, each sized at 224×224 pixels, and spans 1,000 classes. We used a standard approach to select the validation set, randomly sampling 5,000 images from the training sets of CIFAR (Krizhevsky et al., 2009), and 200,000 images from the training set of ImageNet (Deng et al., 2009).

Backbone Architecture and Implementation details: Our approach can be easily integrated into any early exit network. In our experiments, we focus on two of the most representative early exit architectures: MSDNet (Huang et al., 2017b) and RANet (Yang et al., 2020). These architectures are widely used as backbones in the field of dynamic neural networks due to their effectiveness and versatility. We employed a 7-exit MSDNet and a 6-exit RANet on the CIFAR-100 dataset, and a 5-exit MSDNet on the ImageNet dataset.

5.1. Main results

Results on CIFAR-100: We evaluate our method on the CIFAR-100 dataset with a 7-classifier MSDNet. We present the 'Anytime Prediction' setting, showing each classifier's accuracy and its corresponding FLOPs, a standard metric for evaluating computational cost, as detailed in Table 1. The results show that our Bayesian updating method significantly improves performance. By incorporating conformal prediction (CP) and utilizing softmax values, we achieved more accurate likelihood estimates, leading to further performance gains.

We also demonstrated the potential of the Bayesian updating method when accurate likelihoods are available. The results from the Oracle likelihood experiments clearly show the significant performance improvements that Bayesian updating can achieve, highlighting its strong potential in dynamic early exiting networks.

We also evaluated simple voting methods, including standard voting and weighted voting, where weights were based on the model's performance on the validation set. Despite its simplicity and limited use of information, the voting method performs well in practical deep learning applications and serves as a strong alternative when the accuracy of likelihoods limits the performance of Bayesian updating.

Results on ImageNet:

Table 2. **Anytime prediction results** of a 5-exit MSDNet on ImageNet

Exit	1	2	3	4	5
Params ($\times 10^6$)	4.24	8.77	13.07	16.75	23.96
FLOPs ($\times 10^9$)	0.34	0.69	1.01	1.25	1.36
MSDNet	58.78	66.51	69.63	70.74	72.05
Bayes	58.78	63.83	68.26	70.29	71.63
Bayes + CP	58.78	65.27	68.66	70.71	71.84
Bayes + Oracle	58.78	68.94	71.79	73.63	74.61
Bayes + CP + Oracle	58.78	68.89	72.05	74.31	75.11
Logit Voting	58.78	66.72	70.10	71.42	72.32
Softmax Voting	58.78	66.03	69.51	70.96	72.02

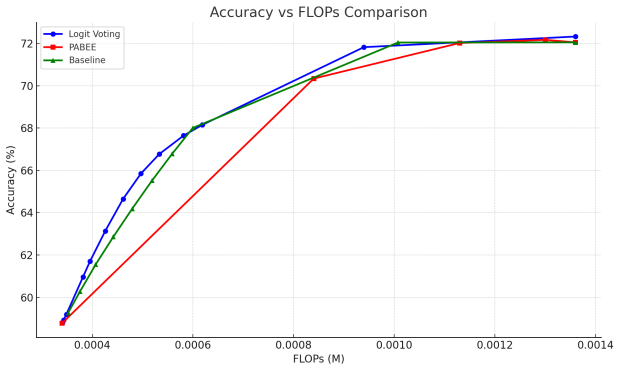


Figure 4. **Dynamic inference** of a 5-exit MSDNet on ImageNet

We tested our method on the large-scale ImageNet dataset, which consists of 1,000 classes, making it a more challenging benchmark, shown in Table 2. For our Bayesian updating method, a higher number of classes increases the difficulty of obtaining reliable likelihood estimates through sampling. Although the ImageNet validation set contains a substantial amount of data, there are only 200 samples per class. Deriving a distribution across 1,000 classes using just 200 samples per class is particularly challenging.

We also evaluate accuracy under varying computational budgets in the dynamic inference setting in Fig 4. Compared to PABEE, our method does not rely on strict output consistency across layers, which allows it to maintain a better accuracy-efficiency trade-off across a wide range of FLOPs.

5.2. Synthetic Experiments

We conducted extensive synthetic experiments to study how different factors impact the performance of Bayesian updating in practical applications. Using a Dirichlet distribution, we generated output distributions for each classifier and label, creating a confusion matrix where the correct label has a higher probability than others.

The generated confusion matrix was used as the true likelihood for the Bayesian updating process. Next, we sampled and counted from the true likelihood to simulate real-world scenarios, replicating the process of practical usage. We performed Bayesian updating with both the true distributions from the classifiers and the sampled empirical distributions. By calculating the error intervals of the posterior distributions at each step, we studied how various factors affect Bayesian updating.

Simulation Settings on CIFAR-100 and ImageNet: Our Bayesian updating method performed less effectively on ImageNet compared to CIFAR-100. To investigate this further, we conducted synthetic experiments, shown in Fig 3, to simulate the task scales of the two datasets, including the number of classes and the number of samples per class. For CIFAR-100, which has 100 classes, we sampled the likelihood 50 times per class. Similarly, for ImageNet with 1,000 classes, we also performed 200 samples per class. We reported the 90% confidence interval, selecting values within the 5%-95% error range. Our synthetic experiments, which simulated the task scale of ImageNet, show that having 1,000 classes with only 50 samples per class caused a large gap between the empirical and true distributions. Even after several steps of Bayesian updating, the error confidence interval remained wide, making the updates unstable.

6. Conclusion and future works

In this paper we studied the performance of two different multi-agent aggregation strategies in dynamic neural networks: a voting based approach, and a technique based on Bayesian updates. We find that theoretically, if we had access to the true likelihood of classifiers, a Bayesian update based approach would perform better than voting. We believe this paper introduces a new and exciting challenge for the multi-agent community, with great potential impact in Computer Vision and Machine Learning.

References

Angelopoulos, A. N. and Bates, S. A gentle introduction to conformal prediction and distribution-free uncertainty quantification. *arXiv preprint arXiv:2107.07511*, 2021.

Angelopoulos, A. N., Bates, S., Jordan, M., and Malik, J. Uncertainty sets for image classifiers using conformal prediction. In *International Conference on Learning Representations*.

Bolukbasi, T., Wang, J., Dekel, O., and Saligrama, V. Adaptive neural networks for efficient inference. In *International Conference on Machine Learning*, pp. 527–536. PMLR, 2017.

Chen, X., Dai, H., Li, Y., Gao, X., and Song, L. Learning to stop while learning to predict. In *International conference on machine learning*, pp. 1520–1530. PMLR, 2020.

Deng, J., Dong, W., Socher, R., Li, L.-J., Li, K., and Fei-Fei, L. Imagenet: A large-scale hierarchical image database. In *2009 IEEE conference on computer vision and pattern recognition*, pp. 248–255. Ieee, 2009.

Ding, T., Angelopoulos, A., Bates, S., Jordan, M., and Tibshirani, R. J. Class-conditional conformal prediction with many classes. *Advances in neural information processing systems*, 36:64555–64576, 2023.

do Carmo Alves, M. A., Varma, A., Elkhatib, Y., and Marcolino, L. S. It is among us: Identifying adversaries in ad-hoc domains using q-valued bayesian estimations. In *AAMAS*, pp. 472–480, 2024.

Dosovitskiy, A. An image is worth 16x16 words: Transformers for image recognition at scale. *arXiv preprint arXiv:2010.11929*, 2020.

Han, S., Mao, H., and Dally, W. J. Deep compression: Compressing deep neural networks with pruning, trained quantization and huffman coding. *arXiv preprint arXiv:1510.00149*, 2015.

Han, Y., Huang, G., Song, S., Yang, L., Wang, H., and Wang, Y. Dynamic neural networks: A survey. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 44 (11):7436–7456, 2021.

Han, Y., Pu, Y., Lai, Z., Wang, C., Song, S., Cao, J., Huang, W., Deng, C., and Huang, G. Learning to weight samples for dynamic early-exiting networks. In *European conference on computer vision*, pp. 362–378. Springer, 2022.

He, K., Zhang, X., Ren, S., and Sun, J. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 770–778, 2016.

He, Y., Liu, P., Wang, Z., Hu, Z., and Yang, Y. Filter pruning via geometric median for deep convolutional neural networks acceleration. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 4340–4349, 2019.

Howard, A. G., Zhu, M., Chen, B., Kalenichenko, D., Wang, W., Weyand, T., Andreetto, M., and Adam, H. Mobilenets: Efficient convolutional neural networks for mobile vision applications. *arXiv preprint arXiv:1704.04861*, 2017.

Huang, C., Lucey, S., and Ramanan, D. Learning policies for adaptive tracking with deep feature cascades. In *Proceedings of the IEEE international conference on computer vision*, pp. 105–114, 2017a.

Huang, G., Chen, D., Li, T., Wu, F., Van Der Maaten, L., and Weinberger, K. Q. Multi-scale dense networks for resource efficient image classification. *arXiv preprint arXiv:1703.09844*, 2017b.

Huang, G., Liu, Z., Pleiss, G., Van Der Maaten, L., and Weinberger, K. Q. Convolutional networks with dense connectivity. *IEEE transactions on pattern analysis and machine intelligence*, 44(12):8704–8716, 2019.

Hubara, I., Courbariaux, M., Soudry, D., El-Yaniv, R., and Bengio, Y. Binarized neural networks. *Advances in neural information processing systems*, 29, 2016.

Jung, S., Son, C., Lee, S., Son, J., Han, J.-J., Kwak, Y., Hwang, S. J., and Choi, C. Learning to quantize deep networks by optimizing quantization intervals with task loss. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 4350–4359, 2019.

Kaya, Y., Hong, S., and Dumitras, T. Shallow-deep networks: Understanding and mitigating network overthinking. In *International conference on machine learning*, pp. 3301–3310. PMLR, 2019.

Krizhevsky, A., Hinton, G., et al. Learning multiple layers of features from tiny images. 2009.

Krizhevsky, A., Sutskever, I., and Hinton, G. E. Imagenet classification with deep convolutional neural networks. *Advances in neural information processing systems*, 25, 2012.

LeCun, Y., Denker, J., and Solla, S. Optimal brain damage. *Advances in neural information processing systems*, 2, 1989.

List, C. and Goodin, R. E. Epistemic democracy: Generalizing the condorcet jury theorem. 2001.

Marcolino, L. S. *Three fundamental pillars of decision-centered teamwork*. University of Southern California, 2016.

Marcolino, L. S., Jiang, A. X., and Tambe, M. Multi-agent team formation: Diversity beats strength? In *IJCAI*, volume 13, 2013.

Marcolino, L. S., Xu, H., Jiang, A. X., Tambe, M., and Bowring, E. Give a hard problem to a diverse team: Exploring large action spaces. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 28, 2014.

Sandler, M., Howard, A., Zhu, M., Zhmoginov, A., and Chen, L.-C. Mobilenetv2: Inverted residuals and linear bottlenecks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 4510–4520, 2018.

Shafer, G. and Vovk, V. A tutorial on conformal prediction. *Journal of Machine Learning Research*, 9(3), 2008.

Sun, Y., Li, J., and Xu, X. Meta-gf: Training dynamic-depth neural networks harmoniously. In *European Conference on Computer Vision*, pp. 691–708. Springer, 2022.

Teerapittayanon, S., McDanel, B., and Kung, H.-T. Branchynet: Fast inference via early exiting from deep neural networks. In *2016 23rd international conference on pattern recognition (ICPR)*, pp. 2464–2469. IEEE, 2016.

Vovk, V., Gammerman, A., and Shafer, G. *Algorithmic learning in a random world*, volume 29. Springer, 2005.

Wołczyk, M., Wójcik, B., Bałazy, K., Podolak, I. T., Tabor, J., Śmieja, M., and Trzcinski, T. Zero time waste: Recycling predictions in early exit neural networks. *Advances in Neural Information Processing Systems*, 34: 2516–2528, 2021.

Yang, L., Han, Y., Chen, X., Song, S., Dai, J., and Huang, G. Resolution adaptive networks for efficient inference. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 2369–2378, 2020.

Yang, L., Jiang, H., Cai, R., Wang, Y., Song, S., Huang, G., and Tian, Q. Condensenet v2: Sparse feature reactivation for deep networks. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 3569–3578, 2021.

Zhang, X., Zhou, X., Lin, M., and Sun, J. Shufflenet: An extremely efficient convolutional neural network for mobile devices. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 6848–6856, 2018.

Zhou, W., Xu, C., Ge, T., McAuley, J., Xu, K., and Wei, F. Bert loses patience: Fast and robust inference with early exit. *Advances in Neural Information Processing Systems*, 33:18330–18341, 2020.