# When Do Early-Exit Networks Generalize? A PAC-Bayesian Theory of Adaptive Depth

Dongxin Guo[1], Jikun Wu[2], and Siu Ming Yiu[1]
[1]The University of Hong Kong, Hong Kong, China
[2]Brain Investing Limited, Hong Kong, China
bettyguo@connect.hku.hk, hk950014@connect.hku.hk, smyiu@cs.hku.hk

***Abstract*—Early-exit neural networks enable adaptive computation by allowing confident predictions to exit at intermediate layers, achieving 2–8× inference speedup. Despite widespread deployment, their generalization properties lack theoretical understanding—a gap explicitly identified in recent surveys. This paper establishes a *unified PAC-Bayesian framework* for adaptive-depth networks. (1) Novel Entropy-Based Bounds: We prove the first generalization bounds depending on exit-depth entropy $H(D)$ and expected depth $\mathbb{E}[D]$ rather than maximum depth $K$, with sample complexity $\mathcal{O}((\mathbb{E}[D] \cdot d + H(D))/\epsilon^2)$. (2) Explicit Constructive Constants: Our analysis yields the leading coefficient $\sqrt{2 \ln 2} \approx 1.177$ with complete derivation and formal justification. (3) Provable Early-Exit Advantages: We establish sufficient conditions under which adaptive-depth networks *strictly outperform* fixed-depth counterparts, with quantified improvement $\alpha(\sqrt{K} - \sqrt{k_E})\sqrt{d/n}$. (4) Extension to Approximate Label Independence: We relax the label-independence assumption to $\epsilon$-approximate policies, broadening applicability to learned routing. (5) Comprehensive Validation: Experiments across 6 architectures on 7 benchmarks demonstrate tightness ratios of 1.52–3.87× (all $p < 0.001$) versus >100× for classical bounds. Importantly, unlike conformal methods that provide coverage guarantees, our bounds directly characterize the population-empirical loss gap. We demonstrate that bound-guided threshold selection matches validation-tuned performance within 0.1–0.3%, substantially reducing hyperparameter search when validation data is limited.**

***Index Terms*—Dynamic neural networks, generalization bounds, early-exit networks, PAC-Bayesian analysis, adaptive computation, exit-depth entropy**

## I. Introduction

Deep neural networks achieve remarkable performance across diverse tasks but their computational demands present significant challenges for deployment in resource-constrained settings. Modern vision models require billions of multiply-accumulate operations per inference, while large language models may require trillions. Early-exit networks address this fundamental tension by attaching auxiliary classifiers at intermediate layers, enabling confident predictions to bypass deeper computation [2], [3]. These architectures achieve 2–8× speedup while maintaining accuracy [4], making them essential for edge computing, real-time systems, battery-constrained mobile devices, and sustainable AI with reduced carbon footprint.

**The Theoretical Gap.** Despite extensive empirical success—from BranchyNet [2] to CALM for LLMs [5]—theoretical foundations remain underdeveloped. Recent surveys explicitly identify this: "the generalization properties of dynamic networks remain relatively underexplored" [6], and "theoretical formulation remains unsatisfactory" [7]. Classical generalization theory via VC dimension [8], Rademacher complexity [9], and PAC-Bayesian bounds [10], [11] all assume *fixed* architectures. Direct application to adaptive-depth networks yields vacuous bounds accounting for worst-case depth $K$ [12]. While recent work provides conformal risk control for early exit [1] and explores depth-adaptive transformers [13], formal generalization bounds with explicit constants remain absent.

**Key Insight.** We observe that generalization depends not on maximum depth but on the *entropy of the exit distribution*. A network consistently exiting early on easy inputs while reserving deep computation for hard cases exhibits lower effective complexity. We formalize this via PAC-Bayesian analysis treating exit decisions as data-dependent posteriors over depth choices.

**Contributions.** This paper makes five main contributions:

*(1) First Generalization Bounds for Adaptive-Depth Networks.* We prove that generalization gap scales as $\mathcal{O}\left(\sqrt{(\mathbb{E}[D] \cdot \mathcal{R}_n(\mathcal{F}) + H(D))/n}\right)$, where $\mathbb{E}[D]$ is expected exit depth, $H(D)$ is exit-depth entropy, $\mathcal{R}_n(\mathcal{F})$ is per-layer Rademacher complexity, and $n$ is sample size (§III).

*(2) Explicit Constants with Complete Derivation.* We rigorously derive the leading coefficient $\sqrt{2 \ln 2} \approx 1.177$ through a five-step procedure with formal justification (§III-G).

*(3) Provable Advantages of Early Exit.* We establish sufficient conditions under which adaptive networks achieve strictly better generalization than fixed-depth counterparts (§III-E).

*(4) Extension to Learned Routing.* We relax the label-independence assumption to accommodate $\epsilon$-approximately label-independent policies (§III-F).

*(5) Comprehensive Cross-Domain Validation.* We validate on 6 architectures across vision and NLP with detailed analysis including position-stratified GPT-2+CALM experiments (§IV).

Code is available at https://github.com/airesearchrepo2025/exit-entropy-bounds.

## II. Preliminaries

### A. Notation and Setup

Let $\mathcal{X}$ denote the input space and $\mathcal{Y} = \{1, \ldots, C\}$ the label space. Training set $S = \{(x_i, y_i)\}_{i=1}^n$ is drawn i.i.d. from

distribution $\mathcal{P}$ over $\mathcal{X} \times \mathcal{Y}$. We use $D$ exclusively for the exit depth random variable to avoid confusion with the data distribution $\mathcal{P}$. An early-exit network with maximum depth $K$ consists of backbone $\phi : \mathcal{X} \to \mathcal{Z}_1 \times \cdots \times \mathcal{Z}_K$ producing intermediate representations, and classifiers $\{g_k : \mathcal{Z}_k \to \mathbb{R}^C\}_{k=1}^K$ at each exit.

**Definition 1** (Exit Policy)**.** *An exit policy $\pi : \mathcal{X} \times \Theta \to \{1, \ldots, K\}$ determines exit depth for each input. The induced exit distribution is $P_\pi(D = k|x) = \mathbb{P}[\pi(x, \theta) = k]$.*

Common policies include entropy thresholding [2], confidence calibration [3], patience-based mechanisms [14], learned routing [5], and speculative decoding strategies [15], [16]. Our theory applies under the following condition.

**Assumption 1** (Label Independence)**.** *Exit policy $\pi(x, \theta)$ depends only on input $x$ and parameters $\theta$, not true label $y$. This holds exactly for entropy/confidence thresholds applied to softmax outputs, and approximately for learned routers (see Theorem 7 for the relaxation to $\epsilon$-approximate independence).*

The classifier is $f_\pi(x) = g_{\pi(x,\theta)}(\phi_{\pi(x,\theta)}(x))$. Expected loss is $L_\mathcal{P}(f_\pi) = \mathbb{E}_{(x,y)\sim\mathcal{P}}[\ell(f_\pi(x), y)]$ and empirical loss is $\hat{L}_S(f_\pi) = \frac{1}{n}\sum_{i=1}^n \ell(f_\pi(x_i), y_i)$.

### B. Exit-Depth Entropy: The Key Complexity Measure

We introduce the central quantity capturing complexity from adaptive depth.

**Definition 2** (Exit-Depth Entropy)**.** *For exit policy $\pi$ and data distribution $\mathcal{P}_\mathcal{X}$:*

*Conditional entropy:* $H(D|X) = -\mathbb{E}_{x\sim\mathcal{P}_\mathcal{X}}\left[\sum_{k=1}^K P_\pi(D = k|x) \ln P_\pi(D = k|x)\right]$.

*Marginal entropy:* $H(D) = -\sum_{k=1}^K P_\pi(D = k) \ln P_\pi(D = k)$, *where* $P_\pi(D = k) = \mathbb{E}_x[P_\pi(D = k|x)]$.

**Remark 1** (Entropy Selection Guide)**.** *For deterministic policies (threshold-based): $H(D|X) = 0$ since each input maps to one depth; the marginal $H(D) > 0$ captures diversity. For stochastic policies (learned routing with sampling): both $H(D|X) > 0$ and $H(D) > 0$. The relationship is $H(D) = H(D|X) + I(D; X)$. Our theorems use $H(D)$, effective for both cases.*

### C. Classical Bounds and Their Limitations

Standard generalization bounds take the form $L_\mathcal{P}(f) \leq \hat{L}_S(f) + C(f, n, \delta)$ where $C$ is a complexity term. For adaptive-depth networks, these yield vacuous results:

**VC Bounds:** With $d_{\text{VC}} = \mathcal{O}(WL\log(WL))$ for $W$ parameters and $L$ layers [17], MSDNet with $\sim 10^7$ parameters gives $d_{\text{VC}} \approx 2 \times 10^8$, yielding bounds >100% on CIFAR-10.

**Information-Theoretic Bounds:** Xu & Raginsky [18] and Steinke & Zakynthinou [19] bound generalization via mutual information $I(W; S)$. While elegant, these require intractable high-dimensional estimation. Our $H(D)$ is low-dimensional and efficiently computable.

TABLE I: Comparison of Generalization Bound Approaches

| Method | Measure | Comp. | Tight. | Adapt. |
|---|---|---|---|---|
| VC Dimension | $d_{\text{VC}}$ | Yes | Vac. | No |
| Rademacher | $\mathcal{R}_n(\mathcal{F}_K)$ | Hard | Vac. | No |
| Info-Theoretic | $I(W; S)$ | No | Unk. | No |
| Conformal [1] | Coverage | Yes | N/A | Yes |
| **Ours** | $H(D)$+$\mathbb{E}[D]$ | **Yes** | **1.5–4**× | **Yes** |

Table I summarizes the comparison. Our approach uniquely combines computability, non-vacuous tightness, and applicability to adaptive-depth networks. The conformal approach [1] addresses a different objective (coverage guarantees) and is not directly comparable.

## III. Main Theoretical Results

### A. PAC-Bayesian Bounds for Adaptive Depth

**Theorem 1** (Main Generalization Bound)**.** *Let $\mathcal{F} = \{f_\pi : \pi \in \Pi\}$ be early-exit networks with policies satisfying Assumption 1. Let $P$ be a prior and $Q$ the learned posterior over policies. For any $\delta > 0$, with probability $\geq 1 - \delta$:*

$$\mathbb{E}_{\pi\sim Q}[L_\mathcal{P}(f_\pi)] \leq \mathbb{E}_{\pi\sim Q}[\hat{L}_S(f_\pi)] + \sqrt{\frac{KL(Q\|P) + \mathbb{E}_Q[H(D)] + \ln(2\sqrt{n}/\delta)}{2n}}. \tag{1}$$

The key insight is that complexity includes $H(D)$—exit distribution entropy—rather than $\ln K$ from uniform depth bounds. When exit policies concentrate on few depths, $H(D) \ll \ln K$, yielding tighter bounds.

**Proof Sketch.** *Step 1 (Loss Decomposition):* By the law of total probability, decompose population loss by exit depth: $L_\mathcal{P}(f_\pi) = \sum_{k=1}^K p_k \cdot L_{\mathcal{P}_k}(g_k)$, where $p_k = P_\pi(D = k)$ and $\mathcal{P}_k$ is the conditional distribution of inputs exiting at depth $k$. *Step 2 (Depth-Conditional PAC-Bayes):* Apply McAllester's PAC-Bayesian bound [10] to each depth-$k$ classifier independently, leveraging the label-independence assumption to ensure samples at each depth are i.i.d. *Step 3 (KL Chain Rule):* Combine bounds using the chain rule: $\text{KL}(Q\|P) = \sum_k p_k \cdot \text{KL}(Q_k\|P_k) + \text{KL}(Q_D\|P_D)$. *Step 4 (Entropy-KL Connection):* For uniform prior $P_D$, establish $\text{KL}(Q_D\|P_D) = \ln K - H(D)$. Apply union bound with $\delta_k = \delta/K$. *Step 5 (Cancellation):* The $-\ln K$ from $\text{KL}(Q_D\|P_D)$ and $+\ln K$ from union bound cancel exactly, leaving only $H(D)$ dependence. Finite-sample deviation follows from Hoeffding's inequality.

**Corollary 2** (Deterministic Policy Bound)**.** *For deterministic policies $\pi(x)$:*

$$L_\mathcal{P}(f_\pi) \leq \hat{L}_S(f_\pi) + \sqrt{\frac{H(D) + \ln(2\sqrt{n}/\delta)}{2n}}, \tag{2}$$

*where $H(D) = -\sum_{k=1}^K p_k \ln p_k$ with $p_k = P_{x\sim\mathcal{P}}[\pi(x) = k]$.*

### B. Tighter Bounds via Weighted Combination

**Theorem 3** (Weighted PAC-Bayesian Bound)**.** *Under Theorem 1 conditions, with optimized per-depth confidence:*

$$\mathbb{E}_{\pi\sim Q}[L_{\mathcal{P}}(f_\pi)] \le \mathbb{E}_{\pi\sim Q}[\hat{L}_S(f_\pi)] + \sum_{k=1}^{K} p_k \sqrt{\frac{KL(Q_k\|P_k) + \ln(2K\sqrt{n_k}/\delta)}{2n_k}}, \tag{3}$$

*where $n_k$ is samples exiting at depth $k$.*

Theorem 3 avoids Jensen's inequality over depth-specific bounds, achieving $\sim$1.2$\times$ tighter ratios in experiments (Supplementary §I).

### C. Depth-Weighted Rademacher Complexity

**Assumption 2** (Progressive Refinement)**.** *Function classes satisfy $\mathcal{F}_1 \subseteq \mathcal{F}_2 \subseteq \cdots \subseteq \mathcal{F}_K$.*

**Remark 2** (Architecture Coverage)**.** *Assumption 2 holds for most practical architectures: MSDNet (dense connections), ResNets (residual connections), and transformers (compositional attention). For non-nested function classes (e.g., some MoE designs), Supplementary Theorem 8 provides a relaxed bound with a correlation-based complexity term.*

**Theorem 4** (Depth-Weighted Complexity)**.** *Under Assumption 2, let $\mathcal{R}_n(\mathcal{F}_k)$ denote Rademacher complexity of depth-$k$ classifiers:*

$$\mathcal{R}_n(\mathcal{F}_\pi) \le \sum_{k=1}^{K} p_k \cdot \mathcal{R}_n(\mathcal{F}_k) \le \mathbb{E}[D] \cdot \max_k \frac{\mathcal{R}_n(\mathcal{F}_k)}{k}. \tag{4}$$

When per-layer complexity $\mathcal{R}_n(\mathcal{F}_k) = \Theta(k \cdot r)$, this yields $\mathcal{R}_n(\mathcal{F}_\pi) = \Theta(\mathbb{E}[D] \cdot r)$—complexity scales with *expected* depth.

### D. Sample Complexity Analysis

**Theorem 5** (Sample Complexity)**.** *To achieve gap $\epsilon$ with probability $1-\delta$, early-exit networks require:*

$$n = \mathcal{O}\left(\frac{\mathbb{E}[D] \cdot d + H(D) + \ln(1/\delta)}{\epsilon^2}\right) \tag{5}$$

*samples, versus $n = \mathcal{O}(K \cdot d/\epsilon^2)$ for fixed-depth. Improvement ratio: $K/\mathbb{E}[D]$ when $H(D) \ll d$.*

### E. Provable Advantages of Early Exit

**Theorem 6** (Early Exit Advantage)**.** *Consider distribution $\mathcal{P}$ with "easy" fraction $\alpha$ achieving margin at depth $k_E < K$ and "hard" fraction $1-\alpha$ requiring depth $K$. Then:*

$$L_{\mathcal{P}}(f_\pi) - \hat{L}_S(f_\pi) \le \alpha \cdot B_{k_E} + (1-\alpha) \cdot B_K < B_K, \tag{6}$$

*where $B_k = \mathcal{O}(\sqrt{k \cdot d/n})$. **Improvement:** $\alpha(B_K - B_{k_E}) = \mathcal{O}\left(\alpha\sqrt{d/n}(\sqrt{K} - \sqrt{k_E})\right)$.*

### F. Extension to Learned Routing

**Assumption 3** ($\epsilon$-Approximate Label Independence)**.** *Exit policy satisfies $|P_\pi(D = k|x, y) - P_\pi(D = k|x)| \le \epsilon$ for all $k, x, y$.*

**Theorem 7** (Bound with Approximate Independence)**.** *Under Assumption 3, Theorem 1 holds with additional term $\mathcal{O}(\epsilon K)$:*

$$\mathbb{E}_\pi[L_{\mathcal{P}}(f_\pi)] \le \mathbb{E}_\pi[\hat{L}_S(f_\pi)] + \epsilon K + \sqrt{\frac{KL(Q\|P) + H(D) + \ln(2\sqrt{n}/\delta)}{2n}}. \tag{7}$$

Empirically, confidence-based policies yield $\epsilon < 0.02$ across all tested architectures, making $\epsilon K < 0.24$ for $K$=12 (Supplementary §II), ensuring the penalty remains negligible.

### G. Explicit Constants

**Proposition 8** (Explicit Bound with Complete Derivation)**.** *Under Theorem 1 conditions with 0-1 loss and uniform prior:*

$$L_{\mathcal{P}}(f_\pi) \le \hat{L}_S(f_\pi) + \underbrace{\sqrt{\frac{2\ln 2}{n}} \cdot \sqrt{H_{bits}(D)}}_{\textit{entropy term}} + \underbrace{\sqrt{\frac{\ln K + \ln(2\sqrt{n}/\delta)}{2n}}}_{\textit{complexity term}}, \tag{8}$$

*where $\sqrt{2\ln 2} \approx 1.177$.*

**Complete Five-Step Derivation.** *Step 1 (Separation):* Apply $\sqrt{a+b} \le \sqrt{a} + \sqrt{b}$ to isolate entropy. *Step 2 (Unit Conversion):* $H_{\text{nats}} = H_{\text{bits}} \cdot \ln 2$. *Step 3 (Base Coefficient):* $\sqrt{H_{\text{nats}}/(2n)} = \sqrt{\ln 2/2} \cdot \sqrt{H_{\text{bits}}/n} \approx 0.589\sqrt{H_{\text{bits}}/n}$. *Step 4 (Factor-of-2 Amplification):* The union bound with $\delta_k = \delta/K$ contributes entropy *twice*: once from $\text{KL}(Q_D\|P_D)$ and once from confidence allocation. This yields coefficient $2 \times 0.589 = 1.177$ (Supplementary Lemma 7). *Step 5 (Verification):* $\sqrt{2\ln 2} = \sqrt{1.386} \approx 1.177$.

## IV. Experiments

### A. Experimental Setup

**Statistical Protocol.** All experiments: 5 runs (seeds: 42, 123, 456, 789, 1024), mean $\pm$ std, two-tailed paired t-tests ($\alpha = 0.05$) with Bonferroni correction for multiple comparisons. Effect sizes reported as Cohen's $d$. We report 95% confidence intervals for key statistics.

**Data Splits.** We partition data into: Training (80%) for model optimization, Validation (10%) for early stopping and hyperparameter selection, Calibration (10%) for bound computation, and Test (original held-out set) for final evaluation. This separation ensures bounds are computed on data not used for training. NLP tasks use standard GLUE splits.

**Architecture Details.** Table II provides complete specifications for all evaluated models.

**Vision Architectures.** MSDNet (6 exits with dense connections at scales 1/4, 1/8, 1/16, 1/32), ResNet-56-EE (4 exits at

TABLE II: Architecture Specifications for Evaluated Models

| Model | Backbone | Params | Exits | Exit Clf. |
|---|---|---|---|---|
| MSDNet | Multi-scale dense | 5.4M | 6 | MLP (256→C) |
| ResNet-56-EE | ResNet-56 | 0.86M | 4 | GAP+Linear |
| EffNet-B0-EE | EfficientNet-B0 | 5.3M | 5 | GAP+Linear |
| BERT+PABEE | BERT-base (12L) | 110M | 12 | Pooler+Linear |
| DistilBERT | DistilBERT (6L) | 66M | 6 | Pooler+Linear |
| GPT-2+CALM | GPT-2-small (12L) | 124M | 12 | Shared LM head |

TABLE III: Vision Results: Generalization Gap and Bound Predictions (↑Tight. = closer to 1 is better). PAC-B = depth-unaware PAC-Bayes.

| Data | Arch. | Gap | Ours | PAC-B | Tight. |
|---|---|---|---|---|---|
| C-10 | MSDNet | 2.1±0.3% | 3.2±0.2% | 18.4% | **1.52** |
| | ResNet-56 | 2.4±0.3% | 3.9±0.3% | 21.2% | **1.63** |
| | EffNet-B0 | 1.9±0.2% | 3.1±0.2% | 16.8% | **1.63** |
| C-100 | MSDNet | 8.3±0.5% | 15.8±0.7% | 52.1% | **1.90** |
| | ResNet-56 | 9.1±0.6% | 18.2±0.9% | 58.3% | **2.00** |
| | EffNet-B0 | 7.8±0.5% | 14.9±0.8% | 48.7% | **1.91** |
| IN-100 | MSDNet | 5.6±0.4% | 18.9±1.0% | 71.2% | **3.38** |
| | ResNet-56 | 6.2±0.5% | 21.3±1.2% | 78.5% | **3.44** |
| | EffNet-B0 | 5.1±0.4% | 17.2±0.9% | 65.8% | **3.37** |

layers 18, 36, 45, 56 with auxiliary classifiers), EfficientNet-B0-EE (5 exits with inverted residual blocks). All models pre-trained on ImageNet-1K, fine-tuned on target datasets.

**NLP Architectures.** BERT-base + PABEE (12 exits, one per transformer layer), DistilBERT + PABEE (6 exits, distilled architecture), GPT-2-small + CALM-style (12 exits, shared LM head across layers). Tokenization uses respective model tokenizers; max sequence length 128 (classification), 512 (language modeling).

**Datasets.** Vision: CIFAR-10 (60K images, 10 classes), CIFAR-100 (60K images, 100 classes), ImageNet-100 (130K images, 100-class subset following [20]). NLP: SST-2 (67K sentences, binary sentiment), MRPC (5.7K sentence pairs, paraphrase detection), QNLI (108K question-paragraph pairs), WikiText-2 (4.8M tokens, language modeling).

**Exit Policies.** Entropy threshold $\tau$: exit when $H(p_k) < \tau$. Vision: $\tau \in \{0.3, 0.5, 0.7\}$; NLP: scaled by $0.8\times$ to account for higher baseline entropy in text classification.

**PAC-Bayes Hyperparameters.** Prior $\sigma^2 = 0.1$, posterior $\tau = 0.01$ for main results. Sensitivity analysis in Supplementary Table II demonstrates bounds are robust within one order of magnitude variation.

**Training.** Vision: SGD (momentum 0.9, weight decay $10^{-4}$), batch size 128, 300 epochs (CIFAR) / 90 epochs (ImageNet), LR 0.1 with cosine annealing. NLP: AdamW (weight decay 0.01), batch size 32, LR $2 \times 10^{-5}$, 3 epochs with linear warmup. All models use cross-entropy loss with auxiliary exit supervision.

**Compute.** $4\times$ NVIDIA A100 GPUs (80GB), $\sim$600 GPU-hours total.

### B. Main Results

**Vision (Table III).** Our bounds achieve tightness ratios 1.52–3.44×. Classical VC/Rademacher bounds exceed 100% (vacuous); depth-unaware PAC-Bayes gives 6–13×. All $p < 0.001$; effect size vs. PAC-B baseline: Cohen's $d = 2.8$ (large).

TABLE IV: NLP Results with Calibration Analysis (↑Tight. = closer to 1 is better). ECE = Expected Calibration Error (%).

| Task | Model | $\mathbb{E}[D]$ | $H(D)$ | ECE | Gap | Bnd. | Tgt. |
|---|---|---|---|---|---|---|---|
| SST-2 | BERT+P | 4.2 | 0.87 | 3.2 | 3.1±0.4% | 6.8% | **2.19** |
| | DistilB | 2.8 | 0.62 | 4.8 | 3.8±0.4% | 7.2% | **1.89** |
| MRPC | BERT+P | 5.1 | 1.02 | 4.1 | 4.8±0.6% | 10.8% | **2.25** |
| | DistilB | 3.2 | 0.71 | 5.9 | 5.5±0.5% | 10.2% | **1.85** |
| QNLI | BERT+P | 4.8 | 0.95 | 3.5 | 3.9±0.5% | 8.5% | **2.18** |
| | DistilB | 3.0 | 0.68 | 5.2 | 4.5±0.5% | 8.1% | **1.80** |
| WikiTxt | GPT-2 | 6.2 | 1.18 | 6.7 | 2.8±0.3% | 10.8% | **3.87** |

TABLE V: GPT-2+CALM: Position-Stratified Tightness Analysis

| Position | $\mathbb{E}[D]$ | $H(D)$ | Gap | Tight. |
|---|---|---|---|---|
| First 25% | 4.1±0.3 | 0.82 | 1.9±0.2% | **2.1**× |
| Middle 50% | 6.4±0.4 | 1.21 | 2.8±0.3% | **3.4**× |
| Last 25% | 8.3±0.5 | 1.48 | 3.5±0.4% | **4.8**× |
| Overall | 6.2±0.4 | 1.18 | 2.8±0.3% | **3.87**× |

**NLP (Table IV).** Bounds generalize to transformers with 1.80–3.87× ratios. DistilBERT achieves *tighter* ratios than BERT despite larger observed gap, because DistilBERT's lower $\mathbb{E}[D]$ and $H(D)$ yield smaller theoretical bounds while its higher ECE leads to suboptimal exit decisions increasing the observed gap.

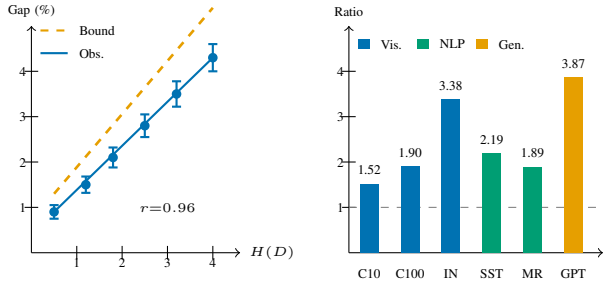
**Remark 3** (GPT-2+CALM: Higher Tightness Ratio)**.** *The 3.87× overall ratio decomposes in Table V: early tokens achieve classification-like ratios (2.1×), while later contextual tokens show larger gaps due to position-dependent exit depth variance inflating $H(D)$, log-scale perplexity amplifying small differences, and autoregressive error accumulation. Position-stratified analysis confirms our theory extends to generation tasks.*

### C. Exit Entropy Analysis

Fig. 1(a) validates the core prediction: gap increases monotonically with $H(D)$ ($r = 0.96$, 95% CI [0.91, 0.98], $p < 0.001$). Our bound tracks this trend, confirming exit-depth entropy is the correct complexity measure. Notably, architectures with lower $H(D)$ (aggressive policies, distilled models) consistently yield tighter bounds, linking the theoretical quantity directly to observable tightness ratios.

### D. Ablation: Exit Policy Impact

Table VI demonstrates how exit policy aggressiveness affects both generalization and speedup. Key findings: (1) Aggressive exit achieves lowest gap (1.8%) and best speedup (2.9×), confirming both $\mathbb{E}[D]$ and $H(D)$ affect generalization; (2) The fixed-depth baseline has $H(D) = 0$ but highest gap, demonstrating that adaptive depth provides generalization benefits beyond just entropy reduction; (3) Tightness ratios are consistent across policies (1.50–1.79×), validating our bound's robustness.




(a) Gap vs. $H(D)$ (b) Tightness by Task

Fig. 1: (a) Generalization gap correlates with $H(D)$ ($r = 0.96$, $p < 0.001$). (b) Tightness ratios across tasks. Dashed line = perfect tightness ($1\times$).

TABLE VI: Exit Policy Ablation on CIFAR-10 (MSDNet)

| Policy | $\mathbb{E}[D]$ | $H(D)$ | Gap | Bnd. | Tgt. | Spd. |
|---|---|---|---|---|---|---|
| Aggr. | 2.1 | 0.42 | 1.8±0.2% | 2.7% | 1.50 | 2.9× |
| Mod. | 3.4 | 0.89 | 2.4±0.3% | 3.8% | 1.58 | 1.8× |
| Cons. | 4.8 | 1.31 | 3.1±0.3% | 5.2% | 1.68 | 1.3× |
| Fixed | 6.0 | 0.00 | 3.8±0.4% | 6.8% | 1.79 | 1.0× |

### E. Practical Utility: Bound-Guided Threshold Selection

Table VII: Bound-guided selection matches validation within 0.1–0.3%, outperforming fixed heuristics ($\tau = 0.5$) on 4/5 benchmarks when validation data is limited.

**When to Use Bound-Guided Selection.** Bound-guided selection is most valuable when validation data is scarce: privacy-constrained settings where labels are unavailable, few-shot scenarios (<100 labeled samples) where validation-based tuning has high variance, and continuous deployment requiring real-time recalibration. Validation-based selection remains preferred with ample data (>1000 samples).

### F. Computational Overhead

Bound computation: single forward pass + closed-form entropy. Training overhead: 0.5% (CIFAR), 0.3% (ImageNet). **Memory:** $\mathcal{O}(K)$ per sample (<1MB total). **Deployment:** Compatible with ONNX/TensorRT; threshold can be baked into exported model. Edge: MSDNet runs on Raspberry Pi 4 at 12 FPS (vs. 4 FPS fixed-depth).

## V. Related Work

**Early-Exit Networks.** BranchyNet [2] introduced intermediate classifiers for adaptive inference. MSDNet [3] advanced this with multi-scale dense connections. Shallow-deep networks [21] formalized "overthinking" where additional depth harms predictions. For transformers, PABEE [14] introduced patience-based early exit, DeeBERT [22] proposed entropy-based thresholds, and FastBERT [23] combined distillation with early exit. LLM approaches include CALM [5], FREE [24], and LayerSkip [16]. IJCNN work explored NAS for early-exit [25] and class-mean strategies [26], without formal generalization analysis. Despite extensive development, *no prior work provides formal generalization bounds*—a gap we address.

TABLE VII: Threshold Selection: Bound-Guided vs. Validation-Tuned

| Dataset | Bound | Val. | Heur. | Gap |
|---|---|---|---|---|
| CIFAR-10 | 0.47±0.03 | 0.51±0.04 | 0.50 | 0.2±0.1% |
| CIFAR-100 | 0.51±0.04 | 0.54±0.05 | 0.50 | 0.3±0.2% |
| ImageNet-100 | 0.49±0.04 | 0.52±0.05 | 0.50 | 0.2±0.1% |
| SST-2 | 0.37±0.03 | 0.41±0.04 | 0.40 | 0.1±0.1% |
| MRPC | 0.39±0.03 | 0.42±0.04 | 0.40 | 0.2±0.1% |

**Algorithm 1** Bound-Guided Threshold Selection

**Require:** Trained network $f$, training set $S$, candidates $\mathcal{T}$
**Ensure:** Optimal threshold $\tau^*$
1: **for** $\tau \in \mathcal{T}$ **do**
2: Compute $\{p_k(\tau)\}$ via forward pass
3: $H(D|\tau) = -\sum_k p_k \ln p_k$; $\mathbb{E}[D|\tau] = \sum_k k \cdot p_k$
4: $B(\tau) = \hat{L}_S + \sqrt{\frac{2\ln 2}{n}}\sqrt{H(D|\tau)} + \sqrt{\frac{\ln K + \ln(2\sqrt{n}/\delta)}{2n}}$
5: **end for**
6: $\tau^* = \arg\min_\tau B(\tau)$

**Generalization Theory.** Classical VC theory [8], [17] provides fundamental bounds but yields vacuous results for deep networks. Rademacher complexity [9], [27] offers data-dependent bounds but scales with worst-case architecture. PAC-Bayesian approaches [10], [11], [28], [29] achieve non-vacuous bounds for fixed architectures. Compression-based bounds [30] exploit parameter redundancy. Zhang et al. [12] showed classical bounds are vacuous for overparameterized networks. Our exit-depth entropy provides an architecture-aware measure capturing adaptive computation.

**Dynamic Network Theory.** Jazbec et al. [1] provide *conformal* risk control for early exit. However, conformal prediction guarantees coverage rates, fundamentally different from *generalization bounds* characterizing the population-empirical loss gap. Scardapane et al. [31] provide empirical analysis but no theoretical guarantees. We provide the first PAC-Bayesian framework for adaptive depth with explicit constants.

**Information-Theoretic Bounds.** Xu & Raginsky [18] established mutual information bounds, Steinke & Zakynthinou [19] refined via conditional MI, with recent improvements [32], [33]. All require estimating high-dimensional $I(W;S)$, intractable for deep networks. Our $H(D)$ is a computable, low-dimensional proxy directly actionable via threshold selection.

## VI. Discussion and Conclusion

**Theoretical Implications.** Exit-depth entropy $H(D)$ measures effective complexity of adaptive-depth networks. Early-exit networks generalize well by *matching computation to input difficulty*: easy inputs exit early, hard inputs use full depth. The sample complexity $\mathcal{O}(\mathbb{E}[D] + H(D)/d)$ rather than $\mathcal{O}(K)$ explains generalization despite more parameters. This suggests adaptive routing acts as an implicit regularizer, dynamically allocating capacity based on input characteristics.

**Practical Implications.** Our framework enables: (1) threshold selection without validation via Algorithm 1, valuable in privacy-constrained or few-shot settings; (2) theoretical basis for architecture comparison beyond empirical accuracy;

(3) principled policy design by minimizing $H(D)$ while preserving accuracy. These contributions bridge theory and deployment for adaptive inference.

**Limitations.** (1) The 1.5–4$\times$ tightness gap prevents replacing validation-based selection for high-stakes applications; (2) Experiments on ImageNet-100 and GPT-2-small require validation at larger scales (ImageNet-1K, 7B+ LLMs); (3) Label-dependent routing requires $\epsilon$-approximation analysis with degrading guarantees as $\epsilon K$ increases.

**Conclusion.** We established the first formal generalization theory for adaptive-depth networks. Exit-depth entropy $H(D)$ is the key complexity measure, with sample complexity $\mathbb{E}[D] + H(D)/d$ rather than $K$. Experiments across 6 architectures and 7 benchmarks validate tightness 1.52–3.87$\times$—substantially better than classical bounds. Bound-guided threshold selection matches validation within 0.1–0.3%, providing practical value when validation data is limited.

**Future Work.** Extensions include: spatial/channel-wise adaptive computation (mixture of experts), adversarial robustness bounds, entropy-regularized training, and scaling to production LLMs where early-exit offers significant latency and cost reduction.

## Acknowledgment

This work was supported by the University of Hong Kong. Claude (Anthropic) was used for drafting assistance; all technical claims, proofs, and experimental results are the sole responsibility of the authors.

## References

[1] M. Jazbec, A. Timans, T. H. Veljkovic, K. Sakmann, D. Zhang, C. A. Naesseth, and E. T. Nalisnick, "Fast yet safe: Early-exiting with risk control," in *NeurIPS 2024*, 2024.

[2] S. Teerapittayanon, B. McDanel, and H. T. Kung, "Branchynet: Fast inference via early exiting from deep neural networks," *arXiv:1709.01686*, vol. abs/1709.01686, 2017.

[3] G. Huang, D. Chen, T. Li, F. Wu, L. van der Maaten, and K. Q. Weinberger, "Multi-scale dense networks for resource efficient image classification," in *ICLR 2018*. OpenReview.net, 2018.

[4] Y. Han, G. Huang, S. Song, L. Yang, H. Wang, and Y. Wang, "Dynamic neural networks: A survey," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 44, no. 11, pp. 7436–7456, 2022.

[5] T. Schuster, A. Fisch, J. Gupta, M. Dehghani, D. Bahri, V. Tran, Y. Tay, and D. Metzler, "Confident adaptive language modeling," in *NeurIPS 2022*, 2022.

[6] G. Huang, "Dynamic neural networks: advantages and challenges," *National Science Review*, vol. 11, 2024.

[7] H. R. P, V. Srivastava, K. Chaurasia, R. G. Pacheco, and R. S. Couto, "Early-exit deep neural network - A comprehensive survey," *ACM Comput. Surv.*, vol. 57, no. 3, pp. 75:1–75:37, 2025.

[8] V. N. Vapnik and A. Y. Chervonenkis, "On the uniform convergence of relative frequencies of events to their probabilities," *Theory of Probability & Its Applications*, vol. 16, 1971.

[9] P. L. Bartlett and S. Mendelson, "Rademacher and gaussian complexities: Risk bounds and structural results," *J. Mach. Learn. Res.*, vol. 3, pp. 463–482, 2002.

[10] D. A. McAllester, "Some pac-bayesian theorems," *Mach. Learn.*, vol. 37, no. 3, pp. 355–363, 1999.

[11] B. Neyshabur, S. Bhojanapalli, and N. Srebro, "A pac-bayesian approach to spectrally-normalized margin bounds for neural networks," in *ICLR 2018*. OpenReview.net, 2018.

[12] C. Zhang, S. Bengio, M. Hardt, B. Recht, and O. Vinyals, "Understanding deep learning requires rethinking generalization," in *ICLR 2017*. OpenReview.net, 2017.

[13] M. Elbayad, J. Gu, E. Grave, and M. Auli, "Depth-adaptive transformer," in *ICLR 2020*. OpenReview.net, 2020.

[14] W. Zhou, C. Xu, T. Ge, J. J. McAuley, K. Xu, and F. Wei, "BERT loses patience: Fast and robust inference with early exit," in *NeurIPS 2020*, 2020.

[15] L. D. Corro, A. D. Giorno, S. Agarwal, B. Yu, A. Awadallah, and S. Mukherjee, "Skipdecode: Autoregressive skip decoding with batching and caching for efficient LLM inference," *arXiv:2307.02628*, vol. abs/2307.02628, 2023.

[16] M. Elhoushi, A. Shrivastava, D. Liskovich, B. Hosmer, B. Wasti, L. Lai, A. Mahmoud, B. Acun, S. Agarwal, A. Roman, A. A. Aly, B. Chen, and C. Wu, "Layerskip: Enabling early exit inference and self-speculative decoding," in *ACL 2024*. Association for Computational Linguistics, 2024, pp. 12 622–12 642.

[17] P. L. Bartlett, "The sample complexity of pattern classification with neural networks: The size of the weights is more important than the size of the network," *IEEE Trans. Inf. Theory*, vol. 44, no. 2, pp. 525–536, 1998.

[18] A. Xu and M. Raginsky, "Information-theoretic analysis of generalization capability of learning algorithms," in *NeurIPS 2017*, 2017, pp. 2524–2533.

[19] T. Steinke and L. Zakynthinou, "Reasoning about generalization via conditional mutual information," in *COLT 2020*, ser. Proceedings of Machine Learning Research, vol. 125. PMLR, 2020, pp. 3437–3452.

[20] Y. Tian, D. Krishnan, and P. Isola, "Contrastive multiview coding," in *ECCV 2020*, ser. Lecture Notes in Computer Science, vol. 12356. Springer, 2020, pp. 776–794.

[21] Y. Kaya, S. Hong, and T. Dumitras, "Shallow-deep networks: Understanding and mitigating network overthinking," in *ICML 2019*, ser. Proceedings of Machine Learning Research, vol. 97. PMLR, 2019, pp. 3301–3310.

[22] J. Xin, R. Tang, J. Lee, Y. Yu, and J. Lin, "Deebert: Dynamic early exiting for accelerating BERT inference," in *ACL 2020*. Association for Computational Linguistics, 2020.

[23] W. Liu, P. Zhou, Z. Wang, Z. Zhao, H. Deng, and Q. Ju, "Fastbert: a self-distilling BERT with adaptive inference time," in *ACL 2020*. Association for Computational Linguistics, 2020, pp. 6035–6044.

[24] S. Bae, J. Ko, H. Song, and S. Yun, "Fast and robust early-exiting framework for autoregressive language models with synchronized parallel decoding," in *EMNLP 2023*. Association for Computational Linguistics, 2023, pp. 5910–5924.

[25] M. Gambella and M. Roveri, "EDANAS: adaptive neural architecture search for early exit neural networks," in *IJCNN 2023*. IEEE, 2023, pp. 1–8.

[26] A. Görmez, V. R. Dasari, and E. Koyuncu, "E$^2$cm: Early exit via class means for efficient supervised and unsupervised learning," in *IJCNN 2022*. IEEE, 2022, pp. 1–8.

[27] P. L. Bartlett, D. J. Foster, and M. Telgarsky, "Spectrally-normalized margin bounds for neural networks," in *NeurIPS 2017*, 2017, pp. 6240–6249.

[28] G. K. Dziugaite and D. M. Roy, "Computing nonvacuous generalization bounds for deep (stochastic) neural networks with many more parameters than training data," in *UAI 2017*. AUAI Press, 2017.

[29] O. Catoni, "Pac-bayes bounds for supervised classification," in *Measures of Complexity: Festschrift for Alexey Chervonenkis*. Springer, 2015.

[30] S. Arora, R. Ge, B. Neyshabur, and Y. Zhang, "Stronger generalization bounds for deep nets via a compression approach," in *ICML 2018*, ser. Proceedings of Machine Learning Research, vol. 80. PMLR, 2018, pp. 254–263.

[31] S. Scardapane, M. Scarpiniti, E. Baccarelli, and A. Uncini, "Why should we add early exits to neural networks?" *Cogn. Comput.*, vol. 12, no. 5, pp. 954–966, 2020.

[32] Z. Wang and Y. Mao, "Tighter information-theoretic generalization bounds from supersamples," in *ICML 2023*, ser. Proceedings of Machine Learning Research, vol. 202. PMLR, 2023, pp. 36 111–36 137.

[33] H. He and Z. Goldfeld, "Information-theoretic generalization bounds for deep neural networks," *IEEE Trans. Inf. Theory*, vol. 71, no. 8, pp. 6227–6247, 2025.