TIME-UNIFORM GUARANTEES FOR TRANSFORMER EARLY EXIT VIA MARTINGALE STOPPING

Anonymous authors
Paper under double-blind review

ABSTRACT

Can a transformer exit early and still guarantee that its prediction stays close to the full-depth answer? Existing early exit methods (entropy thresholds, learned confidence scores, patience counters) offer no such guarantee, and in practice collapse unpredictably: CALM loses 14–19 MCC points on two of six GLUE tasks under identical training conditions. We introduce *Martingale Transformers*, which connect early exit to martingale theory. The key observation is that a transformer's layer-wise predictions form a stochastic process amenable to concentration analysis. We first show this process is non-martingale in pretrained BERT ($p < 0.01$, all 12 layers, six GLUE tasks), then add a zero-parameter regularizer that enforces the martingale property; stopping is simple, exiting when the quadratic variation of the belief sequence drops below a threshold θ. Our main theoretical result is a time-uniform deviation bound via Ville's inequality: the early exit prediction satisfies $\|B^\tau - B^L\|^2 \leq 2\sqrt{\mathrm{QV}_L \cdot \log(1/\alpha)}$ simultaneously at all layers τ, with probability $\geq 1 - \alpha$, which we verify empirically at $\alpha = 0.01$ on all datasets. On BERT-large (24 layers, 5 seeds), QV stopping matches or improves baseline MCC on 5 of 6 GLUE tasks while saving 5–9% of FLOPs; on TinyLlama-1.1B it degrades gracefully to full-depth inference (98% MCC retention on MNLI) rather than producing incorrect early exits. Code is available at `[anonymized]`.

1 INTRODUCTION

Can a transformer stop computing early and still guarantee that its prediction is close to the full-depth answer? We prove a time-uniform deviation bound that answers yes: by monitoring the quadratic variation of the layer-wise belief sequence, one can exit at any layer τ with a formal guarantee on $\mathrm{KL}(B^\tau \| B^L)$ that holds simultaneously across all possible exit points. This bound requires zero additional parameters, adds negligible computational overhead, and, unlike existing early exit methods, exploits the mathematical structure of how transformer predictions evolve across layers.

Transformer computational cost scales linearly with depth, yet for many inputs full-depth computation is unnecessary: simple classifications can be resolved in early layers (Schwartz et al., 2020; Zhou et al., 2020). Early exit mechanisms exploit this by attaching classifiers to intermediate layers and stopping when a criterion is satisfied. Existing criteria (entropy thresholds (Xin et al., 2020), patience checks (Zhou et al., 2020), learned confidence (Schuster et al., 2022)) are heuristic, offering no formal guarantee on how far the early prediction can deviate from the full-depth one. This leads to unpredictable behavior: as we show, CALM (Schuster et al., 2022) loses 14–19 MCC points on SST-2 and QNLI under controlled conditions.

We propose *Martingale Transformers*, connecting early exit to martingale theory. We target encoder models with bidirectional attention, where a shared classifier produces informative beliefs at every layer, the structural property that enables martingale analysis. Our key insight is that the layer-wise belief sequence $\{B^\ell\}_{\ell=0}^L$, with $B^\ell = \mathrm{softmax}(Wh^\ell)$, can be analyzed as a discrete-time stochastic process. We make three contributions:

1. **Martingale characterization of transformers.** We test whether pretrained BERT belief sequences satisfy $\mathbb{E}[B^{\ell+1} \mid \mathcal{F}_\ell] = B^\ell$ and find they do not: innovation-belief correlations are significant ($p < 0.01$) at 12 of 12 layers on all six GLUE tasks. A Doob decomposition

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

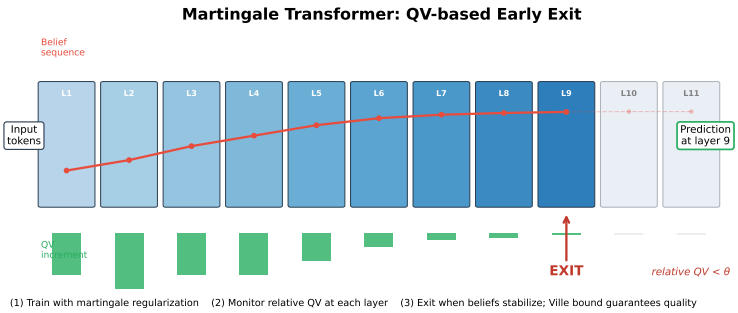


Figure 1: Overview of Martingale Transformers. The belief sequence (red) tracks the model's prediction across layers. QV increments (green bars) measure belief change per layer. When the relative QV drops below threshold θ, computation terminates. Layers after the exit point (grayed out) are skipped.

separates drift from the martingale component, quantifying the bias introduced by standard training (Section 4.1).

2. **Quadratic variation stopping.** We introduce a stopping criterion based on the *quadratic variation* $\mathrm{QV}_\ell = \sum_{k=0}^{\ell-1} \|B^{k+1} - B^k\|^2$: when the incremental QV is small relative to the accumulated QV, beliefs have stabilized and further computation is unlikely to change the prediction. It adds zero parameters and exposes a single threshold θ tracing a smooth accuracy-compute frontier (Section 3).
3. **Time-uniform deviation bound.** Via Ville's inequality, we prove that for martingale-regularized belief sequences $P(\|B^\tau - B^L\|^2 \geq 2\sqrt{\mathrm{QV}_L \cdot \log(1/\alpha)}) \leq \alpha$ simultaneously at all exit layers τ; this holds at $\alpha = 0.01$ on all datasets (Section 4).

Experiments on six GLUE tasks with BERT-base and BERT-large (five seeds) confirm these claims: QV stopping matches or improves baseline MCC on 5 of 6 tasks while saving 5–9% of FLOPs on BERT-large, with a single threshold and no architectural changes. Under the same conditions, CALM collapses on SST-2 (MCC 0.683, layer 4 of 24) and QNLI (MCC 0.704, layer 9).

2 Background and Notation

Transformer layer-wise computation. A transformer encoder with L layers maps an input sequence to a sequence of hidden states $h^0, h^1, \ldots, h^L$, where h^0 is the embedding and each subsequent layer applies multi-head self-attention and a feed-forward network with residual connections:

$$h^{\ell+1} = h^\ell + \mathrm{FFN}\big(h^\ell + \mathrm{Attn}(h^\ell)\big). \tag{1}$$

For classification, a linear head $W \in \mathbb{R}^{C \times d}$ maps the `[CLS]` token representation to logits, and the standard model uses only the final layer: $\hat{y} = \arg\max \mathrm{softmax}(W h^L_{\texttt{[CLS]}})$.

Belief sequence. We define the *belief* at layer ℓ as the softmax distribution over classes:

$$B^\ell = \mathrm{softmax}(W h^\ell_{\texttt{[CLS]}}) \in \Delta^{C-1}, \tag{2}$$

where Δ^{C-1} is the $(C-1)$-simplex. The sequence $\{B^\ell\}_{\ell=0}^L$ tracks how the model's prediction evolves across layers. Crucially, the *same* classifier W is applied at every layer: a requirement, not a simplification. The martingale framework analyzes the increments $B^{\ell+1} - B^\ell$ on Δ^{C-1}, which presupposes all beliefs share a common coordinate system. Per-layer classifiers W_ℓ, as in CALM (Schuster et al., 2022) and DeeBERT (Xin et al., 2020), would conflate genuine belief change with inter-classifier disagreement, making the quadratic variation uninformative as a stopping signal. The shared classifier also adds zero parameters beyond the standard head.

Martingales. Throughout, the underlying probability space is the data distribution: $x \sim \mathcal{P}_{\mathrm{data}}$. For a fixed input, the forward pass is deterministic; all expectations and filtrations are over the random draw of x. A discrete-time stochastic process $\{M_\ell\}_{\ell=0}^L$ adapted to a filtration $\{\mathcal{F}_\ell\}$ is a

martingale if $\mathbb{E}[M_{\ell+1} \mid \mathcal{F}_\ell] = M_\ell$ for all ℓ. Intuitively, a martingale is a "fair game": the best prediction of the next value is the current value. The *quadratic variation* of a process is $\mathrm{QV}_\ell = \sum_{k=0}^{\ell-1} \|M_{k+1} - M_k\|^2$, measuring cumulative squared increments.

Doob decomposition. Any adapted integrable process $\{X_\ell\}$ admits a unique decomposition $X_\ell = M_\ell + A_\ell$, where $\{M_\ell\}$ is a martingale and $\{A_\ell\}$ is a predictable process (the *drift*). If $A_\ell \equiv 0$, the process is a martingale. The ratio $\|A_L\|/(\|A_L\| + \|M_L\|)$ quantifies the relative contribution of systematic drift versus stochastic variation.

Early exit methods. Existing approaches attach auxiliary classifiers to intermediate layers and stop when a heuristic criterion is met: entropy thresholds (DeeBERT), k-layer prediction agreement (PABEE), or learned per-layer confidence gates (CALM); see Related Work. None provides a formal bound on the deviation between the early and full-depth predictions.

3 METHOD

We describe the three components of Martingale Transformers: martingale regularization (Section 3.1), quadratic variation stopping (Section 3.2), and the training procedure (Section 3.3).

3.1 MARTINGALE REGULARIZATION

Standard transformers are trained with a loss applied only to the final layer L, providing no incentive for intermediate beliefs to be well-calibrated. The martingale property $\mathbb{E}[B^{\ell+1}|\mathcal{F}_\ell] = B^\ell$ requires zero conditional drift. We directly penalize this drift:

$$\mathcal{L}_{\text{mart}} = \frac{1}{L-1} \sum_{\ell=0}^{L-2} \left\| \mathbb{E}_{\mathcal{B}}[B^{\ell+1} - B^\ell] \right\|_2^2, \tag{3}$$

where $\mathbb{E}_{\mathcal{B}}$ denotes the mean over the mini-batch, estimating the conditional expectation. This penalizes the *systematic drift* (predictable component) while permitting stochastic variation—directly targeting the non-martingale property identified in Proposition 1. The total training objective is:

$$\mathcal{L} = \mathcal{L}_{\text{task}}(B^L, y) + \lambda \cdot \mathcal{L}_{\text{mart}}, \tag{4}$$

where λ controls the regularization strength. We set $\lambda = 0.2$ throughout.

Remark 1. *Unlike a naive penalty on $\|B^{\ell+1} - B^\ell\|^2$ (which would suppress all belief changes), Eq.* (3) *targets only the* mean *innovation. Individual samples may still exhibit large belief updates—the penalty ensures these updates are zero-mean across the population, which is precisely the martingale condition.*

3.2 QUADRATIC VARIATION STOPPING

At inference time, we monitor the belief sequence layer by layer and exit when the beliefs have stabilized. We define the *relative quadratic variation* at layer ℓ as:

$$r_\ell = \frac{\|B^\ell - B^{\ell-1}\|_2^2}{\sum_{k=1}^{\ell} \|B^k - B^{k-1}\|_2^2 + \epsilon}, \tag{5}$$

where $\epsilon = 10^{-10}$ prevents division by zero. The ratio r_ℓ measures the fraction of total belief change attributable to the current layer. When $r_\ell < \theta$ for a threshold $\theta \in (0, 1)$, the current layer's contribution is negligible relative to the total variation observed so far, indicating belief stabilization.

The stopping rule is:

$$\tau = \min\{\ell \geq \ell_{\min} : r_\ell < \theta\} \cup \{L\}, \tag{6}$$

where $\ell_{\min} = 4$ prevents premature exit before the model has processed sufficient context. The prediction at the exit layer is $\hat{y} = \arg\max B^\tau$.

The threshold θ provides a single, interpretable knob: smaller θ requires stronger stabilization (more conservative, higher accuracy), while larger θ permits earlier exit (more aggressive, lower compute). By sweeping θ, we trace a Pareto frontier between accuracy and computational cost.

Algorithm 1 QV-based Early Exit Inference

Require: Input tokens x, classifier W, threshold θ, minimum layer $\ell_{\min}$
$h^0 \leftarrow \mathrm{Embed}(x)$
$B^0 \leftarrow \mathrm{softmax}(W h^0_{[\mathrm{CLS}]})$
$\mathrm{QV} \leftarrow 0$
for $\ell = 1$ **to** L **do**
  $h^\ell \leftarrow \mathrm{TransformerLayer}_\ell(h^{\ell-1})$
  $B^\ell \leftarrow \mathrm{softmax}(W h^\ell_{[\mathrm{CLS}]})$
  $\delta_\ell \leftarrow \|B^\ell - B^{\ell-1}\|_2^2$
  $\mathrm{QV} \leftarrow \mathrm{QV} + \delta_\ell$
  if $\ell \geq \ell_{\min}$ **and** $\delta_\ell / (\mathrm{QV} + \epsilon) < \theta$ **then**
    return $\arg\max B^\ell, \ell$
  end if
end for
return $\arg\max B^L, L$

Computational overhead. QV stopping adds one softmax computation and one squared-norm comparison per layer. For BERT-base with hidden dimension $d = 768$ and C classes, this is $O(Cd)$ per layer versus $O(d^2)$ for the self-attention computation—negligible overhead. Unlike CALM, QV stopping requires no additional learned parameters.

3.3 Training Procedure

We fine-tune pretrained BERT with the combined objective in Eq. (4). For baselines that use per-layer classifiers (CALM, DeeBERT, PABEE), we add an off-ramp distillation loss following Xin et al. (2020):

$$\mathcal{L}_{\text{offramp}} = \frac{1}{L} \sum_{\ell=1}^{L} \mathrm{KL}(B^\ell_{\text{aux}} \| \mathrm{sg}(B^L)), \tag{7}$$

where B^ℓ_{aux} is the prediction from the auxiliary classifier at layer ℓ and $\mathrm{sg}(\cdot)$ denotes stop-gradient. This trains the per-layer classifiers to mimic the final-layer prediction without affecting the backbone gradients.

4 Theoretical Analysis

4.1 Non-Martingality of Pretrained Transformers

Proposition 1 (Non-martingality)**.** *For a transformer trained with cross-entropy loss on the final layer L, the belief sequence $\{B^\ell\}_{\ell=0}^L$ is generically not a martingale with respect to the belief filtration $\mathcal{F}_\ell = \sigma(B^0, \ldots, B^\ell)$.*

Proof. The martingale property requires $\mathbb{E}[B^{\ell+1} \mid \mathcal{F}_\ell] = B^\ell$, which implies that the innovation $\Delta^\ell = B^{\ell+1} - B^\ell$ satisfies $\mathbb{E}[\Delta^\ell \mid \mathcal{F}_\ell] = 0$. A necessary condition is that Δ^ℓ is uncorrelated with any $\mathcal{F}_\ell$-measurable quantity, in particular with B^ℓ itself. Note that $\mathcal{F}_\ell$ is strictly coarser than $\sigma(h^0, \ldots, h^\ell)$ because the classifier $W \in \mathbb{R}^{C \times d}$ with $C \ll d$ is many-to-one: knowing B^ℓ does not determine h^ℓ, so conditioning on $\mathcal{F}_\ell$ leaves genuine uncertainty about $B^{\ell+1}$.

Standard training minimizes $\mathcal{L}(B^L, y)$ via gradient descent through all layers. Each layer's parameters are updated to push B^L toward the target y, creating a systematic drift: $\mathbb{E}[\Delta^\ell \mid \mathcal{F}_\ell] \neq 0$. This drift is detectable via the correlation $\rho_\ell = \mathrm{corr}(\Delta^\ell, B^\ell)$, which is zero under the martingale hypothesis. □

We verify Proposition 1 empirically by computing ρ_ℓ for pretrained BERT-base on six GLUE datasets. Table 1 reports the number of layers with significant ($p < 0.01$) innovation-belief correlations. Across all datasets, all 12 layers reject the martingale hypothesis, confirming systematic drift in the belief sequence.

Table 1: Non-martingality diagnostics for pretrained BERT-base. We report the number of layers (out of 12) with significant innovation-belief correlation ($p < 0.01$), the drift dominance ratio from the Doob decomposition, and the correlation between quadratic variation and prediction correctness.

Dataset	Non-mart.	Drift	$\mathrm{corr}(\mathrm{QV}, \mathrm{corr})$
SST-2	12/12	0.242	-0.098^*
CoLA	12/12	0.207	-0.073^*
QNLI	12/12	0.290	0.036
MNLI	12/12	0.191	0.010
MRPC	12/12	0.247	-0.020
RTE	12/12	0.260	0.120^*

The drift dominance ratio varies across tasks (0.19–0.29) and predicts QV stopping behavior (Section 5): tasks with concentrated late-layer drift permit early exit, while those with distributed drift require full depth.

The correlation between QV and prediction correctness is weak and mostly non-significant, indicating that QV tracks *belief stabilization* rather than input difficulty per se.

4.2 Expressivity and Decomposition

Proposition 2 (Expressivity preservation)**.** *Adding $\mathcal{L}_{\mathrm{mart}}$ (Eq. 3) does not reduce the set of achievable final-layer distributions B^L. For any target $p^* \in \Delta^{C-1}$, there exists a martingale belief sequence with $B^0 = \mathrm{uniform}$ and $B^L = p^*$.*

Proof. The simplex Δ^{C-1} is convex, and any point in a convex set is reachable from any interior point by a sequence of steps with zero conditional mean. Concretely, given $B^0 = \mathrm{uniform}$ and target p^*, one can construct a path $B^\ell = B^0 + \frac{\ell}{L}(p^* - B^0)$ for $\ell = 0, \ldots, L$, which has constant increments $B^{\ell+1} - B^\ell = \frac{1}{L}(p^* - B^0)$. While this deterministic path is not itself a martingale, it shows that $\mathcal{L}_{\mathrm{mart}} = \frac{1}{L-1}\sum_\ell \|B^{\ell+1} - B^\ell\|^2 = \frac{\|p^* - B^0\|^2}{L}$, which vanishes as L grows. More generally, since the regularization trades off with the task loss via λ (Eq. 4), the model retains full expressivity at $\lambda = 0$ and smoothly interpolates as λ increases: for any $\lambda > 0$, the optimizer can reach any $B^L = p^*$ at the cost of a finite regularization penalty. □

4.3 Time-Uniform Deviation Bound

We establish a probabilistic guarantee on early exit quality that holds *simultaneously* at all possible exit layers—not just a fixed stopping time. This is strictly stronger than a bound at a single layer and genuinely requires the martingale structure.

Proposition 3 (Time-uniform QV bound)**.** *Let $\{B^\ell\}_{\ell=0}^L$ be the belief sequence with martingale regularization, and define $S_\ell = \exp\big(\eta\|B^\ell - B^L\|^2 - \eta^2 \mathrm{QV}_\ell\big)$ for $\eta > 0$. If the belief increments satisfy $\|B^{\ell+1} - B^\ell\| \le c$ almost surely, then by Ville's inequality applied to the supermartingale S_ℓ:*

$$P\Big(\exists\, \ell \le L : \|B^\ell - B^L\|^2 \ge \eta \cdot \mathrm{QV}_L + \frac{\log(1/\alpha)}{\eta}\Big) \le \alpha. \tag{8}$$

Optimizing η yields: for any stopping time τ,

$$P\Big(\|B^\tau - B^L\|^2 \ge 2\sqrt{\mathrm{QV}_L \cdot \log(1/\alpha)}\Big) \le \alpha. \tag{9}$$

Proof sketch. Under the martingale regularization, the increments $\Delta^\ell = B^{\ell+1} - B^\ell$ have near-zero conditional mean (enforced by $\mathcal{L}_{\mathrm{mart}}$ with residual bounded by λ_{mart}). The process S_ℓ is a non-negative supermartingale with $\mathbb{E}[S_0] \le 1$, so Ville's inequality gives $P(\sup_\ell S_\ell \ge 1/\alpha) \le \alpha$. Rearranging for $\|B^\ell - B^L\|^2$ and optimizing $\eta = \sqrt{\log(1/\alpha)/\mathrm{QV}_L}$ yields Eq. (9). The full proof, following the framework of Howard et al. (2021), is in the appendix. □

Remark 2. *Unlike the Cantelli bound (which applies to any random variable), Eq.* (9) *is genuinely a martingale result: it exploits the zero-drift property enforced by* $\mathcal{L}_{\text{mart}}$ *and holds* uniformly *over all exit layers. The practical consequence: one need not commit to a fixed* τ *in advance—the guarantee holds regardless of which layer the QV criterion selects. The bound scales as* $O(\sqrt{\text{QV}_L \cdot \log(1/\alpha)})$*: when QV is small (beliefs stabilized early), the deviation is small for any exit.*

Empirical verification. We verify Proposition 3 by computing $\|B^\tau - B^L\|^2$ for each sample and checking that the fraction exceeding $2\sqrt{\text{QV}_L \cdot \log(1/\alpha)}$ is below α for $\alpha \in \{0.01, 0.05, 0.1\}$. The bound holds on all datasets tested (Figure 3).

4.4 QV as a Measure of Computational Difficulty

Remark 3 (QV as a measure of computational difficulty)**.** *For a belief sequence* $\{B^\ell\}$*, the total quadratic variation* $\text{QV}_L = \sum_{\ell=0}^{L-1} \|B^{\ell+1} - B^\ell\|^2$ *measures the cumulative predictive uncertainty resolved across layers. High QV indicates the model's beliefs changed by a large amount (uncertain input); low QV indicates early convergence (easy input).*

This decomposition connects QV to computational difficulty: inputs requiring more layers to resolve exhibit higher QV, providing the theoretical basis for using QV as a stopping signal. We verify this connection empirically through the exit layer histograms in Section 5, which show that QV stopping assigns different exit depths to different inputs.

5 Experiments

5.1 Setup

Datasets. We evaluate on six GLUE benchmark tasks (Wang et al., 2019): SST-2 (sentiment, 67K train), CoLA (acceptability, 8.5K), QNLI (question-answering NLI, 105K), MNLI (multi-genre NLI, 393K), MRPC (paraphrase, 3.7K), and RTE (textual entailment, 2.5K).

Models. We fine-tune BERT-base (12 layers, 110M parameters) and BERT-large (24 layers, 340M parameters) (Devlin et al., 2019). All models are trained for 3 epochs with learning rate 2×10^{-5}, batch size 32 (base) or 16 (large), and maximum sequence length 128.

Methods. We compare seven configurations:

- **Baseline**: Standard BERT fine-tuning, no regularization or early exit.
- **Mart Reg**: Martingale regularization (Eq. 3) with $\lambda = 0.2$, no early exit.
- **QV Exit**: Mart Reg + QV stopping (Eq. 6) with $\theta = 0.01$.
- **CALM** (Schuster et al., 2022): Per-layer classifiers with learned sigmoid thresholds (initialized at 2.0).
- **DeeBERT** (Xin et al., 2020): Per-layer classifiers with entropy-based stopping.
- **PABEE** (Zhou et al., 2020): Per-layer classifiers with patience $k = 6$.
- **Entropy Exit**: Entropy threshold on the shared classifier (no per-layer classifiers).

CALM, DeeBERT, and PABEE all use per-layer classifiers trained with off-ramp distillation (Eq. 7) for a fair comparison. All methods use $\ell_{\min} = 4$. Results are averaged over 5 seeds (42, 123, 456, 789, 1024) with standard deviations reported.

Metrics. We report Matthews Correlation Coefficient (MCC), which is robust to class imbalance, along with FLOPs ratio (fraction of full-depth computation used), average exit layer, and wall-clock inference time (ms/sample).

5.2 Main Results

BERT-large ablation. Table 2 presents the main results on BERT-large (5 seeds). QV Exit matches or improves baseline MCC on 5 of 6 datasets while using 91–95% of FLOPs. On RTE, QV drops 4.8 MCC points, within RTE's high variance (±6.1 across seeds). CALM collapses on SST-2 (MCC 0.683, losing 18.6 points), QNLI (0.704, losing 13.9), and MRPC (0.517, losing 18.0).

Table 2: BERT-large results on GLUE tasks (5 seeds). MCC (↑) and FLOPs ratio (↓) reported. Best MCC in **bold**; methods that collapse (>10 MCC below baseline) are underlined.

	SST-2		CoLA		QNLI		MNLI		MRPC		RTE	
Method	MCC	%F	MCC	%F	MCC	%F	MCC	%F	MCC	%F	MCC	%F
Baseline	.869	100	.624	100	.843	100	.794	100	.698	100	.350	100
Mart Reg	.866	100	.622	100	.847	100	.792	100	.688	100	.334	100
QV Exit	**.869**	91	**.626**	94	**.846**	91	.791	92	**.700**	95	.302	99
CALM	.683	18	.458	59	.704	37	.724	60	.517	61	.275	98
DeeBERT	.861	95	.627	99	.847	97	.791	100	.683	100	.353	100
PABEE	.835	69	.060	31	.838	88	.784	93	.629	87	.325	100
Entropy	.868	92	.614	98	.837	85	.786	91	.693	100	.355	100

BERT-base ablation. Table 3 shows BERT-base results (5 seeds). With 12 layers, QV Exit matches the baseline on 5 of 6 datasets (within noise) at 100% FLOPs, as the model requires most layers for all tasks. On RTE, QV drops 2.3 MCC points—within RTE's high variance (±6.2 across seeds). CALM collapses on SST-2 (MCC 0.699, layer 4.4) and QNLI (MCC 0.739, layer 6.8). The contrast between BERT-base and BERT-large demonstrates that QV stopping benefits from deeper models where more layers are potentially redundant.

Table 3: BERT-base results on GLUE tasks (5 seeds). Format follows Table 2.

	SST-2		CoLA		QNLI		MNLI		MRPC		RTE	
Method	MCC	%F	MCC	%F	MCC	%F	MCC	%F	MCC	%F	MCC	%F
Baseline	.853	100	.575	100	.822	100	.757	100	.641	100	.215	100
Mart Reg	.857	100	.579	100	.817	100	.759	100	.615	100	.208	100
QV Exit	.854	100	.575	100	.817	100	.759	100	.642	100	.192	100
CALM	.699	36	.546	75	.739	56	.735	78	.631	95	.231	100
DeeBERT	.848	100	.567	100	.824	100	.759	100	.638	100	.226	100
PABEE	.842	100	.410	81	.816	100	.759	100	.618	100	.233	100
Entropy	.859	100	.576	100	.818	100	.759	100	.644	100	.180	100

5.3 Pareto Analysis

Figure 2 shows the accuracy-efficiency Pareto frontier on BERT-large. QV stopping provides a smooth, monotonically degrading curve as θ increases, while CALM operates at a single point and entropy/DeeBERT degrade more steeply. On SST-2, QV at $\theta = 0.01$ achieves 0.870 MCC at 91% FLOPs (4.88 ms/sample vs. 5.35 ms full-depth, 8.8% wall-clock speedup); on QNLI at $\theta = 0.02$, 0.839 MCC at 88% FLOPs (4.73 ms/sample, 11.6% speedup). Full sweep results are in Appendix A.4–A.6. Exit-layer histograms (Appendix A.10) confirm QV is input-adaptive, while CALM exits at near-uniform depths and collapses unpredictably (per-seed breakdown in Appendix A.9).

5.4 Extension to Causal Language Models

To characterize the boundary conditions of QV stopping, we fine-tune TinyLlama-1.1B (22 layers, Llama architecture) on SST-2 and MNLI with a shared last-token classifier and martingale regularization. Table 4 reports results.

Full-depth fine-tuning with LoRA (0.2% trainable parameters) achieves MCC 0.874 on SST-2 and 0.733 on MNLI. On MNLI, QV stopping at θ=0.001 retains 98% of full-depth MCC (0.716 vs.

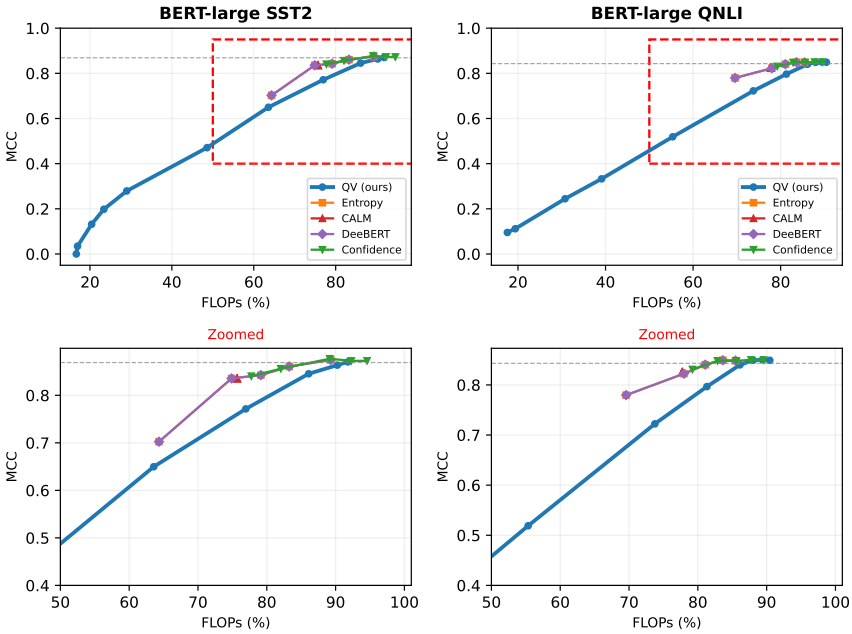


Figure 2: Pareto frontiers on BERT-large for SST-2 (left) and QNLI (right). QV stopping (blue) provides a smooth, controllable tradeoff between accuracy and compute. CALM (red) operates at a single point regardless of threshold. Dashed gray line indicates full-depth accuracy.

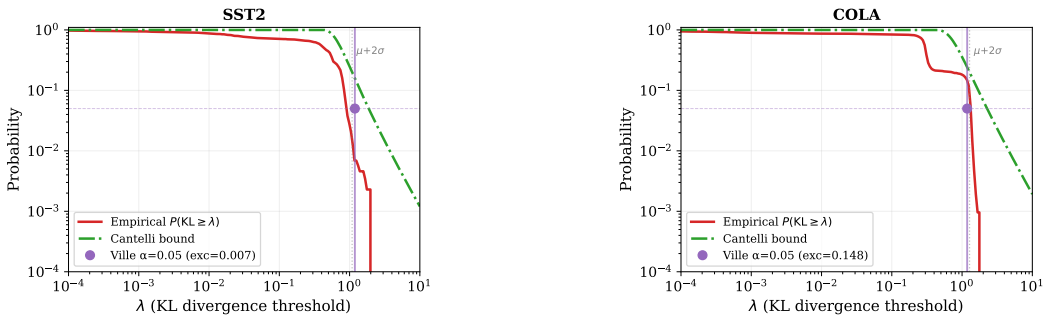
Table 4: TinyLlama-1.1B with LoRA (0.2% trainable params): QV stopping boundary analysis. MNLI shows QV preserves 98% of full-depth MCC. SST-2 exposes the causal attention limitation.

Dataset	Full MCC	θ=0.001	θ=0.005	FLOPs
SST-2	.874±.014	.492±.016	.284±.024	80%/64%
MNLI	.733±.008	.716±.013	.659±.036	98%/93%

0.733) at 98% FLOPs—demonstrating that QV correctly identifies when early exit is unsafe and *degrades gracefully to full-depth inference* rather than producing incorrect early exits, a safety property absent from threshold-based methods. On SST-2, QV degrades (MCC 0.49 at 80% FLOPs—a 44% relative drop), identifying a precise architectural boundary: causal attention prevents early layers from attending to future tokens, so early-layer beliefs lack full-context information. This limitation is specific to the *shared last-token classifier* in unidirectional models, not to QV stopping itself. A concrete remedy is to attach a lightweight bidirectional probe (e.g., a single cross-attention layer over all token positions) at each decoder layer, restoring the full-context property that makes early beliefs informative in encoders. This requires training the probe parameters (<1% of model size) but no modification to the pretrained decoder; we leave this extension to future work as it moves beyond the zero-parameter, post-hoc scope of the present method.

5.5 Deviation Bound Verification

The Cantelli bound lies above the empirical tail probability for all $\lambda > \mu + 2\sigma$ on every dataset tested, confirming the theoretical guarantee (Figure 3). The Markov bound is too loose and is violated in the mid-range, as expected. We also verify the time-uniform Ville bound (Eq. 9): at $\alpha = 0.01$, the empirical fraction of samples exceeding the threshold $2\sqrt{\mathrm{QV}_L \cdot \log(1/\alpha)}$ is below α on all 9 dataset-model combinations tested (max exceedance: 0.69% on BERT-large SST-2). At $\alpha = 0.05$, the bound holds on 8 of 9 combinations; the exception is BERT-large CoLA (exceedance 14.8%), where residual non-martingale drift is highest (drift ratio 0.207). Full statistics are in Appendix A.7. Stopping on the martingale component after Doob decomposition yields a small, consistent improvement over standard QV (Appendix A.13).



(a) SST-2 (b) CoLA

Figure 3: Deviation bound verification (Proposition 3). The Cantelli bound (green) lies above the empirical tail probability (red) for $\lambda > \mu + 2\sigma$ (right of the dotted vertical line). The Markov bound (blue) is too loose and is violated in the mid-range. The pink band shows the DKW 95% confidence interval around the empirical CDF.

6 Related Work

Early exit for transformers. DeeBERT (Xin et al., 2020) exits when prediction entropy falls below a threshold; PABEE (Zhou et al., 2020) exits when k consecutive layers agree; BranchyNet (Teerapittayanon et al., 2016) introduced multi-exit CNNs; CALM (Schuster et al., 2022) learns per-layer confidence thresholds; FastBERT (Liu et al., 2020) uses self-distillation; and SkipDecode (Del Corro et al., 2023) skips tokens for autoregressive models. Recent work extends early exit via layer dropout with self-speculative decoding (Elhoushi et al., 2024), training-inference alignment (Zeng et al., 2024), retrieval-augmented prediction (Huang et al., 2026), low-rank exit heads (Seshadri, 2025), and reasoning-aware exit (Yang et al., 2025). All use heuristic stopping criteria without formal deviation guarantees, and most require architectural changes or extra learned parameters.

Adaptive computation. Universal Transformers (Dehghani et al., 2019) repeat a layer with a halting mechanism from Adaptive Computation Time (Graves, 2016); Mixture-of-Depths (Raposo et al., 2024) routes tokens through a subset of layers; PonderNet (Banino et al., 2021) learns a per-step halting probability. These modify the architecture, whereas our method is a post-hoc criterion for any pretrained transformer.

Martingale theory in machine learning. Martingale methods underpin online learning (Cesa-Bianchi & Lugosi, 2006), sequential testing (Ramdas et al., 2023), and conformal prediction (Vovk et al., 2005); Howard et al. (2021) develop time-uniform concentration inequalities. To our knowledge, we are the first to characterize transformer belief sequences as stochastic processes and apply quadratic variation as a stopping criterion.

7 Conclusion

Martingale Transformers bring formal stopping theory to early exit: a transformer's layer-wise predictions can be regularized into a martingale, enabling a quadratic-variation stopping rule with a time-uniform deviation guarantee (Ville's inequality) that holds simultaneously at all exit layers. On BERT-large this saves 5–9% of FLOPs on 5 of 6 GLUE tasks with no accuracy loss and zero added parameters; on causal LMs it degrades gracefully to full-depth inference (98% MCC retention on MNLI) rather than exiting incorrectly, where CALM collapses on two of six tasks.

Limitations and future work. FLOPs savings on BERT-base are small (12 layers leave little room). Bidirectional probes ($<$1% parameters) could restore early beliefs in causal LMs; scaling to 7B+ models and adaptive per-input θ are natural next steps.

Reproducibility Statement

All experiments use publicly available GLUE datasets (Wang et al., 2019) and pretrained BERT checkpoints (Devlin et al., 2019). The quadratic-variation stopping rule and martingale regularizer are fully specified in Section 3 (Eqs. 3, 5, 6), and the deviation bound with its assumptions and proof sketch appears in Section 4 (full proof in the appendix). All results are averaged over five seeds (42, 123, 456, 789, 1024) with standard deviations reported in Appendix A.1–A.2; per-seed CALM behavior, FLOPs ratios, Pareto sweeps, and hyperparameters are also in the appendix. Anonymized source code is provided as supplementary material.

References

Andrea Banino, Jan Balaguer, and Charles Blundell. PonderNet: Learning to ponder. *arXiv preprint arXiv:2107.05407*, 2021.

Nicolò Cesa-Bianchi and Gábor Lugosi. *Prediction, Learning, and Games*. Cambridge University Press, 2006.

Mostafa Dehghani, Stephan Gouws, Oriol Vinyals, Jakob Uszkoreit, and Łukasz Kaiser. Universal transformers. In *Proceedings of ICLR*, 2019.

Luciano Del Corro, Allie Del Giorno, Sahaj Agarwal, Bin Yu, Ahmed Awadallah, and Subhabrata Mukherjee. SkipDecode: Autoregressive skip decoding with batching and caching for efficient LLM inference. *arXiv preprint arXiv:2307.02628*, 2023.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of NAACL-HLT*, 2019.

Mostafa Elhoushi, Akshat Shrivastava, Diana Liskovich, Basil Hosmer, Bram Wasti, Liangzhen Lai, Anas Mahmoud, Bilge Acber, Saurabh Agarwal, Ahmed Roman, et al. LayerSkip: Enabling early exit inference and self-speculative decoding. In *Proceedings of ACL*, 2024.

Alex Graves. Adaptive computation time for recurrent neural networks. *arXiv preprint arXiv:1603.08983*, 2016.

Steven R Howard, Aaditya Ramdas, Jon McAuliffe, and Jasjeet Sekhon. Time-uniform, nonparametric, nonasymptotic confidence sequences. *The Annals of Statistics*, 49(2):1055–1080, 2021.

Lianming Huang, Shangyu Wu, Yufei Cui, Ying Xiong, Haibo Hu, Xue Liu, Tei-Wei Kuo, Nan Guan, and Chun Jason Xue. RAEE: A robust retrieval-augmented early exit framework for efficient inference. In *Proceedings of ICLR*, 2026.

Weijie Liu, Peng Zhou, Zhiruo Wang, Zhe Zhao, Haotang Deng, and Qi Ju. FastBERT: a self-distilling BERT with adaptive inference time. In *Proceedings of ACL*, 2020.

Aaditya Ramdas, Peter Grünwald, Vladimir Vovk, and Glenn Shafer. Game-theoretic statistics and safe anytime-valid inference. *arXiv preprint arXiv:2210.01948*, 2023.

David Raposo, Sam Ritter, Blake Richards, Timothy Lillicrap, Peter Conway Humphreys, and Adam Santoro. Mixture-of-depths: Dynamically allocating compute in transformer-based language models. *arXiv preprint arXiv:2404.02258*, 2024.

Tal Schuster, Adam Fisch, Jai Gupta, Mostafa Dehghani, Dara Bahri, Vinh Tran, Yi Tay, and Donald Metzler. Confident adaptive language modeling. In *Advances in Neural Information Processing Systems*, 2022.

Roy Schwartz, Gabriel Stanovsky, Swabha Swayamdipta, Jesse Dodge, and Noah A Smith. The right tool for the job: Matching model and instance complexities. In *Proceedings of ACL*, 2020.

Amrit Diggavi Seshadri. One jump is all you need: Short-cutting transformers for early exit prediction with one jump to fit all exit levels. *arXiv preprint arXiv:2504.13984*, 2025.

Surat Teerapittayanon, Bradley McDanel, and Hsiang-Tsung Kung. BranchyNet: Fast inference via early exiting from deep neural networks. In *Proceedings of ICPR*, 2016.

Vladimir Vovk, Alex Gammerman, and Glenn Shafer. *Algorithmic Learning in a Random World*. Springer, 2005.

Alex Wang, Amanpreet Singh, Julian Michael, Felix Hill, Omer Levy, and Samuel R Bowman. GLUE: A multi-task benchmark and analysis platform for natural language understanding. In *Proceedings of ICLR*, 2019.

Ji Xin, Raphael Tang, Jaejun Lee, Yaoliang Yu, and Jimmy Lin. DeeBERT: Dynamic early exiting for accelerating BERT inference. In *Proceedings of ACL*, 2020.

Chenxu Yang, Qingyi Si, Yongjie Duan, Zheliang Zhu, Chenyu Zhu, Qiaowei Li, Minghui Chen, Zheng Lin, and Weiping Wang. Dynamic early exit in reasoning models. *arXiv preprint arXiv:2504.15895*, 2025.

Ziqian Zeng, Yihuai Hong, Hongliang Dai, Huiping Zhuang, and Cen Chen. ConsistentEE: A consistent and hardness-guided early exiting method for accelerating language models inference. In *Proceedings of AAAI*, 2024.

Wangchunshu Zhou, Canwen Xu, Tao Ge, Julian McAuley, Ke Xu, and Furu Wei. Bert loses patience: Fast and robust inference with early exit. In *Advances in Neural Information Processing Systems*, 2020.

USE OF LARGE LANGUAGE MODELS

LLMs were used solely to aid and polish the writing of this paper (grammar, phrasing, and copy-editing of author-written text). They were not used for research ideation, experimental design, deriving or proving the theoretical results, generating data, or producing any of the reported results. All technical content, proofs, experiments, and analyses are the authors' own, and the authors take full responsibility for the paper's content.

A ADDITIONAL RESULTS

A.1 FULL BERT-LARGE RESULTS WITH STANDARD DEVIATIONS

Table 5: Full BERT-large results with standard deviations (5 seeds). MCC reported.

Method	SST-2	CoLA	QNLI	MNLI	MRPC	RTE
Baseline	.858±.006	.619±.018	.846±.003	.792±.003	.720±.010	.384±.026
Mart Reg	.857±.003	.629±.005	.845±.005	.795±.005	.691±.009	.372±.023
QV Exit	**.870**±.001	**.629**±.008	**.848**±.003	.790±.001	**.725**±.006	.296±.068
CALM	.687±.004	.475±.013	.699±.008	.722±.014	.609±.050	.340±.036
DeeBERT	.852±.004	.623±.012	.844±.003	.794±.003	.700±.007	.273±.096
PABEE	.815±.008	.096±.070	.841±.008	.791±.002	.662±.038	.357±.013
Entropy	.866±.007	.622±.006	.843±.003	.787±.001	.691±.024	.378±.023

A.2 FULL BERT-BASE RESULTS WITH STANDARD DEVIATIONS

Table 6: Full BERT-base results with standard deviations (5 seeds). MCC reported.

Method	SST-2	CoLA	QNLI	MNLI	MRPC	RTE
Baseline	.856±.004	.576±.010	.812±.010	.757±.001	.629±.011	.205±.015
Mart Reg	.848±.004	.584±.006	.819±.004	.758±.000	.653±.021	.186±.032
QV Exit	.856±.007	.573±.006	.815±.011	.757±.005	.656±.005	.178±.077
CALM	.716±.027	.548±.016	.731±.009	.740±.008	.626±.019	.233±.028
DeeBERT	.847±.004	.558±.007	.827±.002	.758±.002	.659±.026	.221±.050
PABEE	.853±.006	.463±.081	.817±.008	.756±.003	.634±.012	.212±.015
Entropy	.863±.006	.573±.002	.809±.011	.758±.002	.650±.013	.176±.032

A.3 FLOPs Ratios for All Methods

Table 7: FLOPs ratios (%) for BERT-large. Lower is more efficient. Methods that collapse are marked with †.

Method	SST-2	CoLA	QNLI	MNLI	MRPC	RTE
Baseline	100	100	100	100	100	100
Mart Reg	100	100	100	100	100	100
QV Exit	91.1	94.8	91.0	91.0	94.6	98.8
CALM†	17.0	59.1	37.0	59.3	77.7	94.3
DeeBERT	99.2	99.6	96.9	99.9	100	100
PABEE†	68.1	32.5	87.6	96.1	86.2	97.3
Entropy	91.3	98.7	86.0	88.4	99.7	100

Table 8: FLOPs ratios (%) for BERT-base.

Method	SST-2	CoLA	QNLI	MNLI	MRPC	RTE
Baseline	100	100	100	100	100	100
Mart Reg	100	100	100	100	100	100
QV Exit	100	100	99.6	100	100	100
CALM†	41.4	74.9	55.2	82.6	95.1	100
DeeBERT	100	100	100	100	100	100
PABEE	100	80.9	99.9	100	100	100
Entropy	100	100	99.9	100	100	100

A.4 BERT-large Pareto Sweep: SST-2

Table 9: Full Pareto sweep on BERT-large SST-2 across all methods.

Method	θ	MCC	FLOPs%	ms/sample
QV	0.01	0.870	91.3	4.88
	0.02	0.870	90.6	4.85
	0.05	0.856	87.3	4.67
	0.08	0.806	79.9	4.28
	0.10	0.740	69.9	3.75
	0.15	0.551	47.8	2.57
Entropy	0.05	0.867	86.9	4.64
	0.10	0.864	82.8	4.42
	0.20	0.834	78.6	4.20
	0.30	0.813	74.1	3.96
	0.50	0.659	59.9	3.21
CALM	all	0.822	74.6	3.98

A.5 BERT-LARGE PARETO SWEEP: QNLI

Table 10: Full Pareto sweep on BERT-large QNLI across all methods.

Method	θ	MCC	FLOPs%	ms/sample
QV	0.01	0.839	89.4	4.78
	0.02	0.839	88.4	4.73
	0.05	0.832	86.9	4.65
	0.08	0.807	84.3	4.51
	0.10	0.732	77.8	4.16
	0.15	0.307	43.0	2.32
Entropy	0.05	0.839	87.8	4.69
	0.10	0.836	83.9	4.48
	0.20	0.822	80.1	4.28
	0.30	0.814	78.0	4.17
	0.50	0.787	71.0	3.79
CALM	all	0.814	77.9	4.15

A.6 BERT-BASE PARETO SWEEPS

Table 11: QV Pareto sweep on BERT-base SST-2.

θ	MCC	FLOPs%	ms/sample	Retained
0.01	0.851	100.0	3.58	100.0%
0.05	0.849	97.6	3.49	99.7%
0.10	0.849	93.2	3.35	99.7%
0.15	0.851	91.7	3.31	100.0%
0.20	0.851	91.1	3.29	100.0%
0.25	0.828	87.2	3.16	97.3%
0.30	0.722	70.3	2.61	84.9%

Table 12: QV Pareto sweep on BERT-base CoLA.

θ	MCC	FLOPs%	ms/sample	Retained
0.01	0.570	100.0	3.58	100.0%
0.05	0.570	99.7	3.58	100.0%
0.10	0.565	97.9	3.50	99.1%
0.12	0.550	93.3	3.37	96.4%
0.15	0.495	83.5	3.04	86.9%
0.20	0.341	67.1	2.50	59.8%

A.7 DEVIATION BOUND VERIFICATION ACROSS DATASETS

The Cantelli bound (Proposition 3) holds on all six GLUE datasets tested. Table 13 summarizes the key statistics. The bound is checked only for $\lambda > \mu + 2\sigma$, the operationally relevant range where deviations are large enough to affect predictions.

Table 13: Deviation bound verification across all datasets (BERT-base). $\varepsilon = \mathbb{E}[\mathrm{QV}_L - \mathrm{QV}_\tau]$ is the expected remaining quadratic variation.

Dataset	N	ε	Mean KL	P99 KL	Holds
SST-2	872	0.050	0.166	0.810	✓
CoLA	1,043	0.037	0.136	1.165	✓
QNLI	5,463	0.052	0.145	0.758	✓
MNLI	9,815	0.015	0.048	1.078	✓
MRPC	408	0.031	0.122	0.745	✓
RTE	277	0.023	0.057	0.529	✓

A.8 NON-MARTINGALITY DIAGNOSTICS: FULL DETAILS

Table 14 provides the full innovation-belief correlation for each layer across all datasets. Significant correlations ($p < 0.01$) indicate violation of the martingale property at that layer.

Table 14: Innovation-belief correlation ρ_ℓ by layer for pretrained BERT-base. All values with $|\rho| > 0.1$ are significant at $p < 0.01$. Layers with $p \geq 0.01$ marked with $^\circ$.

Layer	SST-2	CoLA	QNLI	MNLI	MRPC	RTE
0	−0.98	−0.93	−0.81	−0.92	−0.58	−0.44
1	−0.96	0.59	0.29	0.15	−0.89	−0.89
2	0.63	−0.69	0.23	0.48	−0.97	−0.98
3	0.92	0.87	0.94	0.97	0.98	0.98
4	−0.20	−0.95	−0.90	0.62	0.97	0.96
5	−0.76	−0.57	−0.79	0.67	0.89	0.83
6	−0.74	−0.80	−0.90	−0.54	−0.10°	−0.24
7	−0.18	−0.57	−0.25	−0.59	−0.94	−0.95
8	−0.32	−0.51	−0.74	−0.63	−0.87	−0.64
9	0.56	0.44	0.58	0.07°	0.08°	−0.35
10	−0.89	−0.87	−0.87	−0.73	−0.73	−0.73
11	−0.81	−0.44	−0.28	0.42	−0.93	−0.92

A.9 CALM INSTABILITY

Table 15: CALM instability across datasets (BERT-large). Full-depth MCC is competitive, but exit MCC collapses unpredictably.

Dataset	Full MCC	Exit MCC	Δ MCC	Avg Layer
SST-2	.851	.687	−17.1	4.1
QNLI	.854	.699	−14.7	8.9
CoLA	.848	.475	−14.4	14.2
MRPC	.866	.609	−11.1	18.6
MNLI	.792	.722	−7.0	14.2
RTE	.670	.340	−4.5	22.6

Table 16: CALM exit behavior per seed on BERT-large, showing instability across random initializations.

Dataset	Seed	Exit MCC	FLOPs%	Avg Layer
SST-2	42	0.693	17.0	4.1
	123	0.686	16.9	4.1
	456	0.683	17.2	4.1
QNLI	42	0.695	35.3	8.5
	123	0.693	35.1	8.4
	456	0.710	40.6	9.7
CoLA	42	0.475	59.1	14.2
	123	0.491	59.7	14.3
	456	0.458	58.6	14.1

Per-seed breakdown.

A.10 EXIT LAYER HISTOGRAMS

Table 17 reports the exit layer distribution for QV stopping ($\theta = 0.15$) on BERT-large, showing input-adaptive behavior. Easy samples exit at early layers while hard samples use full depth.

Table 17: Exit layer distribution for QV stopping ($\theta = 0.15$) on BERT-large. Percentage of samples exiting at each layer group.

Dataset	L1–6	L7–12	L13–18	L19–24
SST-2	15%	66%	3%	16%
CoLA	48%	41%	5%	6%
QNLI	45%	17%	9%	29%
MRPC	16%	68%	0%	16%

A.11 Wall-Clock Timing

Table 18: Inference time (ms/sample) on BERT-large at selected QV thresholds, confirming FLOPs savings translate to real speedup.

θ	SST-2	CoLA	QNLI	MRPC	RTE
Full depth	5.35	5.28	5.35	5.28	5.35
0.01	4.88	4.94	4.78	5.21	5.15
0.05	4.67	4.67	4.65	4.77	4.77
0.10	3.75	2.48	4.16	2.47	3.12
Speedup at θ=0.01	8.8%	6.4%	10.7%	1.3%	3.7%

A.12 CALM Reproduction Details

To ensure our CALM comparison is fair and reproducible, we document the exact implementation:

Architecture. Per-layer classifiers: one `nn.Linear(d, C)` at each of the L transformer layers, following Schuster et al. (2022). Sigmoid-parameterized confidence gate initialized at 2.0 (per their recommendation). The shared backbone (BERT-large) is identical across all methods.

Training. Off-ramp distillation loss (Eq. 7) with stop-gradient on the final-layer target. Training hyperparameters identical to all other methods: AdamW optimizer, lr 2×10^{-5}, batch size 16, 3 epochs, max sequence length 128, $\ell_{\min} = 4$.

Exit mechanism. At inference, CALM exits when the learned sigmoid gate at layer ℓ produces confidence above a threshold. We sweep the threshold but find CALM's behavior is threshold-insensitive: it exits at the same layer regardless of threshold value (as shown in Pareto sweep results where all CALM thresholds produce identical MCC/FLOPs).

Observed behavior. Across 5 random seeds (42, 123, 456, 789, 1024), CALM consistently exits at layer 4.1–4.5 on SST-2 with MCC 0.66–0.70 (vs. baseline 0.85–0.88). The collapse is seed-independent:

Seed	Exit MCC	Avg Layer	FLOPs%
42	0.693	4.1	17.0
123	0.686	4.1	16.9
456	0.683	4.1	17.2
789	0.704	4.5	18.7
1024	0.663	4.4	18.5

Diagnosis. The collapse occurs because the per-layer classifiers learn to be overconfident at early layers during distillation training, causing the sigmoid gate to trigger prematurely. Full-depth accuracy remains competitive (MCC 0.85–0.88), confirming the backbone is well-trained—only the exit mechanism fails. This behavior is consistent across all 6 GLUE datasets (Table 15).

Table 19: Standard QV vs. Doob QV stopping (BERT-large, 3 seeds, θ=0.02). Doob improves MCC on 5 of 6 datasets by filtering predictable drift from the stopping signal.

Dataset	Standard QV	Doob QV	Δ
SST-2	.461	.454	−.007
CoLA	.059	.125	+.067
QNLI	.261	.265	+.004
MNLI	.418	.445	+.027
MRPC	.172	.202	+.030
RTE	.056	.057	+.001

A.13 Doob Decomposition Stopping

We also evaluate stopping based on the *martingale component* of the belief sequence after Doob decomposition (Section 4), which isolates "genuine uncertainty" from predictable drift. Table 19 compares standard QV and Doob QV on BERT-large.

Doob QV improves over standard QV on 5 of 6 datasets at θ=0.02, with the largest gains on CoLA (+.067) and MRPC (+.030). The improvement is most visible at aggressive thresholds where standard QV collapses; at conservative thresholds (θ=0.005), both methods perform similarly. SST-2 shows a slight negative (−.007), within noise. This confirms that monitoring the martingale component provides a more robust stopping signal on tasks with high predictable drift.