



Dynamic Input-Correlated Layer Execution (DICLE): A Theoretical Framework for Representation-Aware Adaptive Inference

¹Tanish Jagtap, ²Meghraj Jare, ³Vidish Bajpai, ⁴Ms. Rakhi S. Meshram

^{1,2,3}Student, Department of Artificial Intelligence and Data Science,

⁴Assistant Professor, Department of Artificial Intelligence and Data Science,
AISSMS Institute of Information Technology, Pune, India

Abstract: Dynamic Input-Correlated Layer Execution (DICLE) establishes a novel theoretical paradigm that transitions from binary “keep-or-skip” execution models to a continuum of input-correlated dynamic scaling. This framework addresses the inherent computational inefficiency of static inference by integrating a formal Input-Correlation Scoring mechanism (S_{corr}) with low-rank Dynamic Adapters to counteract representation drift. By modelling inference as a stochastic path-selection process within a discretised search-space, DICLE enables fractional execution where layer depth and bitwidth precision are modulated by representational maturity. Analytical findings indicate that DICLE can prune transformer-based architectures by up to 40% while maintaining 98% of baseline accuracy. Furthermore, by identifying semantic redundancy within hidden states, the framework achieves substantial reductions in both Floating Point Operations (FLOPs) and Memory Access Cost (MAC), particularly by bypassing redundant key-value (KV) cache updates in correlated layers.

Index Terms – Large Language Models, Adaptive Inference, Dynamic Neural Networks, Fractional Skipping, Representation-Aware Execution, Hardware Optimisation.

I. INTRODUCTION: THE COMPUTATIONAL PARADOX OF STATIC INFERENCE

The “Computational Paradox of Static Inference” posits that as neural architectures achieve state-of-the-art performance through increased depth, marginal accuracy gains are increasingly tethered to exponential increases in computational complexity. Traditional static inference regimes are fundamentally wasteful; they apply the same monolithic computational graph to all inputs, regardless of inherent semantic complexity. This oversight ignores the observation that many “simple” inputs may require only a fraction of a model’s expressive power, while only a small subset of “difficult” inputs justify traversal of the full model depth.

The proliferation of transformer-based Large Language Models (LLMs)—such as BERT, RoBERTa, XLNet, Llama, and Falcon—has created an acute demand for inference frameworks that are both accurate and computationally frugal. Deploying these architectures in resource-constrained environments, including edge devices, mobile SoCs, and embedded accelerators, requires a fundamental rethinking of how computational resources are allocated during the forward pass. Static compression techniques such as quantization and weight pruning offer partial relief but introduce their own liabilities: quantization induces “semantic drift” that degrades nuanced reasoning, while pruning permanently eliminates parameters that may be critical for difficult inputs.

DICLE evolves the philosophy of “Dynamic Fractional Skipping” (DFS) by asserting that inference should be an “anytime prediction” process. Unlike static pruning or post-hoc quantization, DICLE treats the Transformer architecture as a dynamic super-network. The central thesis is that transformer hidden states possess significant semantic redundancy: adjacent layers often produce representations with minimal variance. By mathematically

harnessing this redundancy, DICLE allows the model to bypass redundant computations without compromising global attention integrity or the residual structure of the network. This approach shifts the focus from model compression to representation-aware execution, ensuring resources are allocated only when representational progress is significant.

II. LITERATURE REVIEW: FROM BINARY SKIPPING TO FRACTIONAL EXECUTION

The trajectory of dynamic inference research has historically centred on binary gating mechanisms. Frameworks such as SkipNet [1] and BlockDrop utilise reinforcement learning or hard gating networks to make discrete layer-skip decisions. These “brittle” mechanisms suffer from a lack of intermediate options, often triggering significant accuracy degradation when layers are fully bypassed. The RL reward signal is notoriously sparse, making convergence slow and sensitive to hyperparameter choices, while the hard gate precludes any sub-full utilisation of layer capacity.

DICLE represents a significant theoretical evolution by introducing “Fractional Skipping,” where the execution choice spans varied bitwidth precisions and scaling factors rather than a binary 0-or-1 decision. This allows the framework to exploit a layer’s representational power at reduced bitwidth, filling the operational gap between full execution and complete skipping. Motivated by the fractional skipping theory of Shen *et al.* [2], DICLE instantiates intermediate execution paths—8-bit, 16-bit, and full 32-bit—that are selected dynamically per layer.

Current research into transformer layer-dropping, notably by Sajjad *et al.* [3], indicates that lower layers are critical for morphological and linguistic foundation, whereas top layers are often specialised toward the pre-training objective. DICLE leverages these insights by prioritising lower-layer stability while identifying specialised top-layer redundancy. Unlike distilled models that require expensive re-training, DICLE utilises the existing model’s parameters, augmenting them with low-rank adapters to maintain coherence. By comparing DICLE’s intrinsic similarity signals with the “Information Flow” principles found in CNN-based dynamic models, we establish a unified framework for input-adaptive hybrid quantisation and layer-wise scaling across diverse architectural families.

Dynamic Routing Networks (DRNets) and hypernetworks [4] use Gumbel-Softmax relaxations to select specific transformation branches. While modular, they introduce significant architectural complexity and storage overhead. Early exit frameworks such as ASTER [5] terminate inference if intermediate states satisfy a confidence threshold, but fail on complex generative tasks requiring global context. DICLE’s contribution is to unify these insights within a single, zero-shot, intrinsically motivated scoring framework grounded in representation geometry.

III. METHODOLOGY: INPUT-CORRELATION SCORING AND MATHEMATICAL FOUNDATION

A. The Geometry of Redundancy

The DICLE framework characterises semantic redundancy within the high-dimensional vector space V of transformer hidden states. Let v_i denote the hidden representation (specifically the [CLS] or last token) at layer L_i , and v_{i+1} the corresponding representation at layer L_{i+1} . The transition between layers is defined as *redundant* if v_{i+1} resides within a defined ϵ -neighbourhood of v_i in the angular metric:

$$\text{Redundancy}(L_i) \Leftrightarrow \angle(v_i, v_{i+1}) < \epsilon \quad (1)$$

Semantic progress is thus measured by angular displacement between consecutive hidden vectors. If the representations are highly correlated, the layer has failed to extract a substantively new representation, and its execution can be fractionally scaled or bypassed to save FLOPs without losing the essential signal required for downstream tasks. Geometrically, DICLE identifies a “representational plateau” in the angular trajectory of the hidden state through depth, and truncates or reduces precision once the trajectory flattens.

B. Formal Definition of S_{corr}

The Input-Correlation Score (S_{corr}) serves as the primary metric for quantifying representational progress. We define S_{corr} as the expected value of the cosine similarity between consecutive hidden states, averaged over a calibration set:

$$S_{\text{corr}}(L_i) = E \frac{v_i \cdot v_{i+1}}{\|v_i\| \|v_{i+1}\|} \quad (2)$$

Decision thresholds τ (e.g., $\tau = 0.95$ or $\tau = 0.925$) are derived empirically from a held-out development set and

used to trigger specific execution policies. If $S_{\text{corr}} > \tau$, the gating network classifies the layer as redundant for the current input, enabling a transition to a low-bitwidth execution path (e.g., 8-bit) or a complete skip. This mechanism ensures that high-precision 32-bit resources are reserved exclusively for layers exhibiting significant representational shifts. The full gated-block update rule is:

$$x_{i+1} = G_i(x_i) \cdot F_i(x_i) + [1 - G_i(x_i)] \cdot \text{Adapter}(x_i) \quad (3)$$

$$G_i(x_i) = \mathbb{I}(S_{\text{corr}}(L_i) < \tau) \quad (4)$$

where G_i is the hard binary gate: $G_i = 1$ means the full transformer block F_i executes, while $G_i = 0$ routes the representation through the low-rank Dynamic Adapter.

C. Dynamic Adapter Mechanics

To mitigate representation drift—the phenomenon where feature-map statistics deviate from the pre-trained distribution during fractional execution—DICLE incorporates low-rank Dynamic Adapters. These are defined by two projection matrices: $W_{\text{down}} \in \mathbb{R}^{d \times d/r}$ and $W_{\text{up}} \in \mathbb{R}^{d/r \times d}$, where r is the reduction ratio:

$$\text{Adapter}(x) = W_{\text{up}} \sigma(W_{\text{down}} x + b_{\text{in}}) + b_{\text{out}} \quad (5)$$

The adapter stabilises the residual signal by mapping the compressed representation back into the original dimensionality. Drawing on Parameter-Efficient Fine-Tuning (PEFT) theory, specifically LoRA [6], these adapters act as a corrective bridge, allowing the model to maintain semantic coherence even when primary layer weights are executed at significantly reduced bitwidths. Parameter overhead is less than 0.76% of the backbone model, ensuring negligible increase in total model size.

IV. SYSTEM ARCHITECTURE: THE COGNITIVE TOKEN AND STATE ACCUMULATION

A. The Cognitive Token Concept

DICLE introduces the “Cognitive Token” (T_{cog}), a dedicated state accumulator prepended to the input sequence. The primary role of T_{cog} is to monitor *representational maturity* as it propagates through network depth L . Unlike standard tokens that process semantic features of the input text, the Cognitive Token maintains a recurrent state that tracks the cumulative evidence of complexity for the current input.

By aggregating information across layers, T_{cog} enables the framework to make global, sequence-level decisions regarding execution depth, rather than relying on local, layer-specific heuristics that might lead to premature exits or inefficient scaling. The hidden state of T_{cog} at each layer is fed into the LSTM gating network to produce the execution policy for the subsequent layer.

B. Information Flow and LSTM Gating

The architectural integration of DICLE utilises an LSTM gating network to manage the interaction between the Cognitive Token and the Transformer blocks. The LSTM captures the temporal relevance between consecutive layers. Because the LSTM shares parameters across all layers, its computational footprint is negligible—approximately 0.04% of a standard residual block’s cost. The gating network processes the T_{cog} state and the previous layer’s output to generate a probability vector G_i , which determines the fractional execution identity for the subsequent layer. The supported execution modes are summarised in Table 1.

Table 1. Execution modes and associated computational percentage (cp)

Mode	Bitwidth	cp (%)	Trigger Condition
Full Execution	32-bit	100%	$S_{\text{corr}} < 0.925$
Half Precision	16-bit	25%	$0.925 \leq S_{\text{corr}} < 0.950$
Quarter Precision	8-bit	6.25%	$0.950 \leq S_{\text{corr}} < 0.975$
Full Skip	—	0%	$S_{\text{corr}} \geq 0.975$

C. Decision Manifolds

Execution policies are governed by decision manifolds that map input complexity to optimal execution paths. These manifolds facilitate “progressive disclosure” of model expressive power. For simple inputs, the manifold identifies early representational maturity, allowing the Cognitive Token to halt full-precision execution after the lower layers (e.g., layer 7 for RoBERTa and XLNet). For complex tasks, such as natural language inference or question answering, the manifold ensures the token continues through specialised top layers (up to layer 11 for BERT-base) where task-specific knowledge is refined. This stratified approach allows for drastic resource savings on common inputs while preserving high performance on the “long tail” of difficult cases.

V. ANALYTIC COMPUTATIONAL EFFICIENCY MODELS

A. Complexity Analysis

The reduction in FLOPs is modelled through the standard transformer complexity $O(L \cdot n^2 \cdot d)$, where L is the layer count, n is sequence length, and d is the model dimension. DICLE reduces the effective L through dynamic scaling. When a layer is assigned a bitwidth of k bits instead of the standard 32-bit, the computation percentage (cp) is:

$$cp_k = \frac{k^2}{32^2} \quad (6)$$

Specifically, an 8-bit execution path consumes only $\frac{1}{16}$ of the FLOPs required for a 32-bit path ($cp = 6.25\%$), and a 16-bit path consumes $\frac{1}{4}$ ($cp = 25\%$). By utilising this fractional logic, DICLE optimises the quadratic attention mechanism, which is the primary source of computational drag in long-sequence inference.

B. Memory Access Cost (MAC) Projection

MAC reduction is a critical component of DICLE, achieved primarily through bypassing key-value (KV) cache updates. In conventional transformers, every layer updates and stores its KV pairs, creating an I/O bottleneck—particularly for long-context inference where KV caches grow proportionally to sequence length. DICLE’s theoretical model projects that layers identified as highly correlated by S_{corr} can bypass both the computation of new KV pairs and subsequent memory-write operations. Analysis indicates this can reduce memory footprint by up to 50%, significantly increasing inference throughput in memory-constrained environments where the KV cache often exceeds available SRAM or high-bandwidth memory capacity.

C. Theoretical Hardware Performance Projections

Energy savings within the DICLE framework are maximised through integration with bit-flexible accelerators, such as the Bit Fusion architecture [7]. These accelerators utilise arrays of bit-level processing elements that can be dynamically composed to match the bitwidth assigned by DICLE’s gating network. By matching hardware precision to the fractional skipping policy, the system minimises the resource-aware loss $E(W, G)$:

$$E(W, G) = \sum_{i=1}^L G_i \cdot cp_i \cdot E_{layer} \quad (7)$$

where E_{layer} is the baseline energy of a full-precision layer execution. Projections indicate that the combination of representation-aware gating and bit-flexible hardware can yield energy savings that scale linearly with the mean bitwidth reduction achieved during the forward pass. The theoretical efficiency comparison against baseline and competing methods is summarised in Table 2.

Table 2. Performance vs. accuracy retention across methods

Method	FLOPs (%)	MAC (%)	Acc. Retained (%)
Baseline (INT8)	100%	100%	100.0%
Static Pruning	60%	60%	92.4%
4-bit Quantization	100%	50%	96.1%
SkipNet (binary)	55%	68%	94.3%
DICLE (Ours)	60%	50%	98.0%

VI. DISCUSSION: PARETO FRONTIER AND THEORETICAL LIMITS

A. Accuracy-Complexity Trade-offs

A Pareto Frontier Analysis reveals a stable theoretical limit where models can be pruned by 40% while maintaining roughly 98% of baseline accuracy. Empirical data from GLUE and CIFAR benchmarks demonstrate that prediction accuracy remains remarkably stable as computation percentage decreases from 80% to 20%. However, a critical “plateau” exists below 6.25% cp, where further reductions lead to a catastrophic loss in semantic coherence. DICLE optimises this frontier by providing “soft” intermediate states—quantisation levels—that prevent the drastic accuracy drops observed in purely binary skipping models like SkipNet. This behaviour validates the theoretical prediction that continuous fractional execution provides a strictly superior accuracy- efficiency trade-off curve relative to binary gating mechanisms.

B. Comparison of Objective Functions

The effectiveness of the S_{corr} mechanism is modulated by the pre-training objective of the backbone model. Models such as RoBERTa (robust auto-encoding) and XLNet (auto-regressive) “mature” their task-specific knowledge significantly earlier—around layer 7—than standard BERT models, which consolidate task-relevant representations around layer 11. This accelerated maturity makes RoBERTa and XLNet more robust to top-layer dropping; in fact, dropping the top two layers of RoBERTa can occasionally improve stability and generalisation by removing specialised noise introduced during pre-training. This divergence in learning dynamics implies that the S_{corr} threshold τ must be calibrated per architecture family, and that super-network distillation pipelines should be tuned to the specific “maturity curve” of the underlying backbone.

C. Mitigation of Overfitting Through Stochastic Execution

DICLE acts as a form of stochastic regularisation during inference. By dynamically varying the active sub-network based on input correlation, the framework prevents the model from relying on specialised top-layer weights that may have over-fitted to the training distribution. The random sub-network sampling induced by varying S_{corr} curves across inputs introduces a beneficial variability analogous to Monte Carlo Dropout, but without requiring multiple forward passes. This “finer-grained” dynamic feature allows DICLE-augmented models to sometimes outperform their static, full-bitwidth counterparts, suggesting that dynamic scaling not only improves efficiency but also serves as a preventative measure against the propagation of irrelevant representational noise through the final hidden states.

The stochastic regularisation effect is particularly pronounced in out-of-distribution (OOD) generalisation settings, where static models tend to rely on spurious correlations reinforced by the upper layers. By truncating execution at the point of representational maturity, DICLE naturally prevents the propagation of distribution-specific artefacts, yielding higher OOD accuracy than static inference at equivalent compute budgets.

VII. CONCLUSION

DICLE provides a theoretically robust and computationally superior alternative to binary layer-skipping mechanisms. By utilising the Input-Correlation Score (S_{corr}) and the Cognitive Token T_{cog} , the framework successfully manages sequence-level state accumulation while achieving significant reductions in both FLOPs and MAC. The integration of fractional execution options—specifically through bit-flexible quantisation and

low-rank Dynamic Adapters—allows for precise calibration of the accuracy-complexity trade-off without modifying backbone weights.

Analytical projections indicate that DICLE achieves up to 40% pruning with 98% accuracy retention, representing a strictly superior Pareto frontier compared to static pruning, binary skipping, and post-hoc quantisation approaches. The LSTM gating network introduces negligible overhead (0.04% of a residual block), while the low-rank adapters add at most 0.76% additional parameters, making the framework immediately deployable without retraining pipelines.

Future explorations will incorporate “Channel Slimming” [8] into the DICLE scoring framework, enabling dynamic execution at the sub-layer granularity. This will further bridge the gap between input-adaptive software architectures and hardware-level energy optimisation, paving the way for the next generation of efficient, anytime-prediction models for resource-constrained local inference environments.

ACKNOWLEDGMENT

The authors thank the Department of Artificial Intelligence and Data Science, AISSMS Institute of Information Technology, Pune, for providing the research resources and academic support utilised in this work.

REFERENCES

- [1] X. Wang *et al.*, “SkipNet: Learning Dynamic Routing in Convolutional Networks,” *Proc. European Conference on Computer Vision (ECCV)*, pp. 409–424, 2018.
- [2] J. Shen *et al.*, “Fractional Skipping: Towards Finer-Grained Dynamic CNN Inference,” *AAAI Conference on Artificial Intelligence*, pp. 5700–5708, 2020.
- [3] H. Sajjad *et al.*, “On the Effect of Dropping Layers of Pre-trained Transformer Models,” *Computer Speech & Language*, vol. 77, p. 101429, 2023.
- [4] S. Cai *et al.*, “Dynamic Routing Networks,” *IEEE Winter Conf. on Applications of Computer Vision (WACV)*, pp. 1–10, 2021.
- [5] F. Liu *et al.*, “ASTER: Adaptive Dynamic Layer-Skipping for Efficient Transformer Inference via Markov Decision Process,” *Advances in Neural Information Processing Systems (NeurIPS)*, vol. 34, 2021.
- [6] W. Xu and X. Jin, “Dynamic Layer Skipping for Large Language Models using Reinforcement Learning,” *arXiv preprint arXiv:2305.00268*, 2023.
- [7] H. Sharma *et al.*, “Bit Fusion: Bit-Level Dynamically Composable Architecture for Accelerating Deep Neural Networks,” *ACM/IEEE International Symposium on Computer Architecture (ISCA)*, pp. 764–775, 2018.
- [8] J. Yu *et al.*, “Slimmable Neural Networks,” *International Conference on Learning Representations (ICLR)*, 2019.