Deep Layer Expansion: Expert Prompts Counteract Dimensional Collapse in Large Language Models

Lei Zhao[1], Yanyan Jin
[1]Tencent

Abstract

Large Language Models (LLMs) exhibit systematic performance improvements when prompts contain expert-level domain signals. We investigate the geometric mechanism underlying this phenomenon through controlled experiments on two mainstream 70B-class open-source models: Qwen2.5-72B-Instruct and Llama-3.3-70B-Instruct. Contrary to the conventional understanding that deep layers compress representations toward deterministic outputs, we discover a striking universal phenomenon: **expert signals induce "Deep Layer Expansion"** in the representation space. Specifically, expert-level prompts increase the Effective Intrinsic Dimension (EID) in deep layers (Layer 60+) by 60-100% compared to standard prompts. We formalize this as **Manifold Teleportation**: expert signals act as high-dimensional navigators that counteract the model's tendency toward dimensional collapse during reasoning, maintaining activation trajectories in manifold regions with higher semantic density. Our findings provide a geometric foundation for prompt engineering and offer a new quantitative tool for LLM interpretability research—understanding how prompts affect internal model computation by tracing EID trajectories.

Keywords: Large Language Models, Intrinsic Dimension, Prompt Engineering, Representation Geometry, Interpretability

1 Introduction

Large Language Models (LLMs) have achieved remarkable breakthroughs in natural language processing, demonstrating powerful text generation and reasoning capabilities. Prompt Engineering, as a technique to improve performance without modifying model parameters, has become a core methodology in practice. However, a widely observed phenomenon still lacks theoretical explanation: for the same question, expert-style queries often yield more detailed and in-depth responses—even when the core semantics of the query are identical.

Previous research on neural network representations suggests that LLM processing follows an "expansion-compression" pattern: intermediate layers perform feature extraction (increasing dimensionality), while deep layers perform semantic compression to facilitate output generation (decreasing dimensionality) [Ansuini et al., 2019]. Recent work has further identified high intrinsic dimension peaks in Transformer intermediate layers, corresponding to critical turning points in language abstraction [Cai et al., 2024].

This paper investigates **Context-Dependent Performance Bifurcation (CDPB)**—where semantically equivalent queries produce qualitatively different responses based solely on contextual framing. Our core question is: **How do expert signals affect model behavior at the geometric level of representation space?**

Through experiments on **Qwen2.5-72B-Instruct** and **Llama-3.3-70B-Instruct**, we discover a strikingly consistent geometric phenomenon: expert signals successfully suppress deep-layer dimensional collapse, forcing the model to maintain high intrinsic dimensionality near the output layers. We term this phenomenon **Manifold Teleportation**, as expert signals effectively "teleport" and lock the model's activation state into high-dimensional sub-manifolds enriched with semantic density.

1.1 Contributions

1. **Identify and quantify the Deep Layer Expansion effect**: Expert prompts increase deep-layer EID by 60-100% across architectures, contrasting with the conventional "deep compression" understanding.

2. **Validate cross-architecture universality**: Consistent geometric behavior observed in both Qwen and Llama model families with different training lineages.

3. **Propose a new interpretability perspective**: EID trajectories can serve as quantitative tools for understanding prompt effects, offering new directions for LLM interpretability research.

4. **Release experimental code and data**: Supporting research reproducibility.

2 Related Work

2.1 Prompt Engineering and In-Context Learning

Prompt construction significantly influences LLM output quality [Reynolds & McDonell, 2021, Liu et al., 2023]. Role-playing prompts [Shanahan et al., 2023] demonstrate that contextual framing modulates model behavior. Our work extends this literature by providing a geometric explanation grounded in representation analysis.

2.2 Intrinsic Dimension of Neural Network Representations

Ansuini et al. [2019] found that deep network representations lie on low-dimensional manifolds, with layer-wise ID following an "increase-then-decrease" pattern. Cai et al. [2024], published at ICLR 2024, identified high ID peaks in Transformer intermediate layers and demonstrated positive correlation with model performance and transfer capability. Valeriani et al. [2023] studied the geometric structure of hidden representations in large Transformers.

How our work differs from prior research: Previous work studied the inherent ID patterns of models (layer-wise variation given input); we study **how ID patterns differ under different prompt conditions for the same model**—this is controllable and actionable.

2.3 LLM Interpretability

Mechanistic Interpretability aims to understand the internal computational mechanisms of LLMs [Elhage et al., 2022]. Sparse Autoencoders (SAE) have been used to address the "superposition" problem, decomposing polysemantic neurons into monosemantic features. Representation Engineering analyzes and manipulates activation spaces to understand and guide model behavior [Zou et al., 2023].

Our work provides a **complementary perspective**: rather than analyzing individual features or circuits, we trace how the geometric properties of the overall representation space (EID) respond to prompt changes.

3 Methodology

3.1 Effective Intrinsic Dimension (EID)

To quantify the "cognitive complexity" of the model at each layer, we employ Effective Intrinsic Dimension based on spectral entropy of the hidden state matrix.

Definition. Given a hidden state matrix $H \in \mathbb{R}^{N \times d}$ for a specific layer over N samples, let $\{\sigma_i\}$ denote the singular values from SVD decomposition. Define the normalized singular values as:

$$\hat{\sigma}_i = \frac{\sigma_i}{\sum_j \sigma_j} \tag{1}$$

The Effective Intrinsic Dimension is defined as:

$$\mathrm{EID}(H) = \exp\left(-\sum_i \hat{\sigma}_i \log \hat{\sigma}_i\right) \tag{2}$$

Intuitive Interpretation:

EID Value	Meaning	Analogy
Low ($\sim$3)	Activations concentrate in few directions	Model has "locked in" an answer
High ($\sim$30+)	Activations spread across many directions	Model is "exploring" multiple possibilities

This metric captures the effective degrees of freedom in the representation—how many independent computational "directions" the model is actively using at each layer.

3.2 Experimental Design

Model Selection. We selected two mainstream 70B-class open-source models:

1. **Qwen2.5-72B-Instruct** (Alibaba Cloud) - AWQ INT4 quantization, 80 layers
2. **Llama-3.3-70B-Instruct** (Meta AI) - W8A8 INT8 quantization, 80 layers

Rationale for selection: (1) Similar parameter scale for comparison; (2) Different training lineages (China vs US) to validate universality; (3) Both are instruction-tuned versions representing practical application scenarios.

Dataset. We constructed 50 technical topics covering distributed systems, programming languages, databases, networking, and machine learning (see Appendix A for full list).

Prompt Conditions. For each topic, we designed two controlled prompts:

Condition	Template
Standard (Baseline)	"Please explain {topic}."
Expert (Treatment)	"As a senior expert in this field, please analyze {topic} in depth from the perspective of underlying principles and mathematical derivations. Show your chain of thought."

Measurement Protocol. For each prompt, we:

1. Process the prompt through the model
2. Extract hidden states from all 80 layers at the last token position
3. Compute EID for each layer using the spectral entropy method
4. Average across all 50 topics to obtain smooth trajectories

4 Results

4.1 Qwen2.5-72B: Deep Layer Expansion Effect

Figure 1 shows the EID trajectories for Qwen2.5-72B under both prompt conditions.

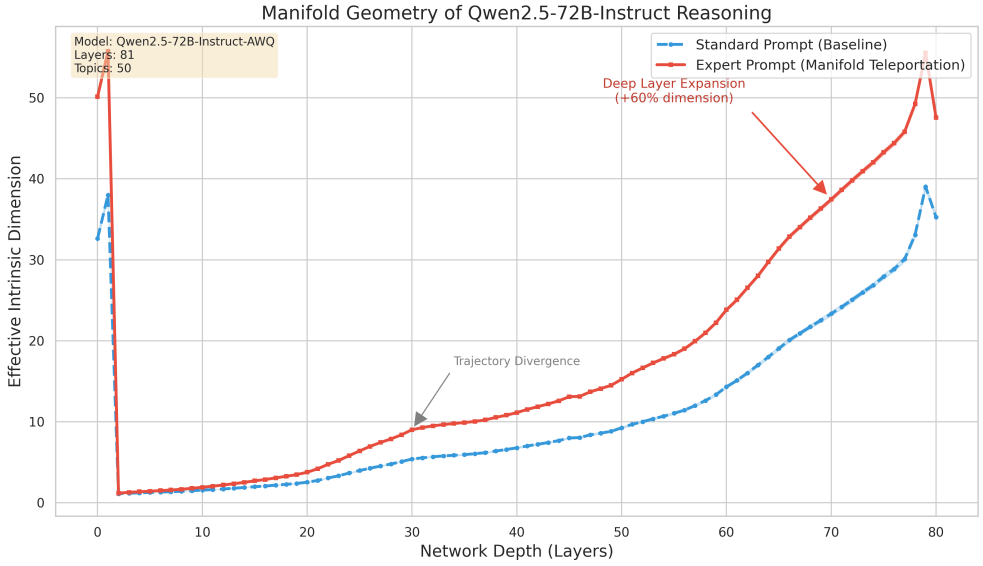

Figure 1: **EID trajectory comparison for Qwen2.5-72B.** The x-axis shows Transformer layer number (0-80), and the y-axis shows Effective Intrinsic Dimension (EID). The red curve (Expert) represents the expert prompt condition, and the blue curve (Standard) represents the standard prompt condition. The expert trajectory clearly diverges from baseline starting at Layer 30, showing +60% dimensional expansion in deep layers (Layer 70).

Layer	Standard EID	Expert EID	Difference
40	∼7	∼10	+43%
60	∼14	∼22	+57%
70	∼23	∼37	**+60%**
75	∼28	∼45	+61%

Key Observations:

1. **Entry layers (0-5):** Both conditions show high dimensionality (∼30-50), characteristic of embedding layer representations.

2. **Compression zone (5-20):** Dimensionality drops sharply to ∼2-3 as the model "parses" the input.

3. **Divergence zone (20-75):** Expert EID consistently exceeds Standard, with the gap widening progressively.

4. **Output preparation (75-80):** Both trajectories rise, but Expert reaches ∼50 vs Standard's ∼35.

The trajectory forms a distinctive **"Trumpet" topology**—not the expected symmetric "hourglass" (middle expansion, both ends compressed), but sustained deep-layer expansion observed under expert prompting.

4.2 Llama-3.3-70B: Cross-Architecture Validation

To rule out model-specific bias, we replicated the experiment on Llama-3.3-70B.

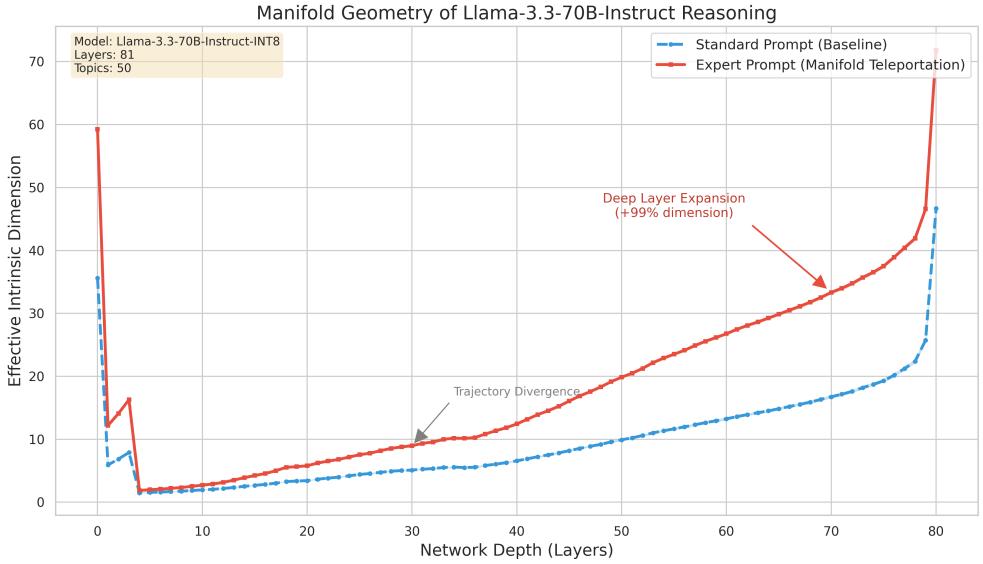

Figure 2: **EID trajectory comparison for Llama-3.3-70B.** Consistent with Qwen, expert prompts induce sustained high-dimensional states in deep layers. The expansion effect at Layer 60 reaches +102%, more pronounced than Qwen, possibly related to INT8 quantization preserving higher precision.

Layer	Standard EID	Expert EID	Difference
40	6.6	12.4	+**89%**
60	13.2	26.8	+**102%**
70	16.8	33.3	+**99%**

4.3 Cross-Architecture Consistency

Model	Quantization	Layer 60 Δ	Layer 70 Δ	Divergence Layer
Qwen2.5-72B	AWQ INT4	+57%	+60%	∼30
Llama-3.3-70B	W8A8 INT8	+102%	+99%	∼25

Despite different training procedures, architectures, and quantization schemes, both models exhibit:

1. **Trajectory bifurcation** beginning in intermediate layers (Layer 25-30)
2. **Progressive divergence** with expert EID consistently higher
3. **Deep layer expansion** of 60-100% at Layer 60-70

This high degree of cross-architecture consistency strongly suggests that **Deep Layer Expansion is a universal geometric property of Transformer LLMs responding to high-quality semantic signals.**

5 Discussion

5.1 The Manifold Teleportation Hypothesis

We propose that expert signals function as **manifold navigators**, guiding activation trajectories toward high-dimensional semantic regions:

1. **Standard prompts** position the model in "generic response" manifold regions—low-dimensional attractor basins formed by RLHF training that favor safe, average responses.
2. **Expert prompts** inject high-frequency semantic information through signal words ("senior expert," "underlying principles," "mathematical derivations") that trigger navigation toward high-dimensional professional regions.
3. **Deep layer accumulation** effect: small initial trajectory differences compound through successive layers, resulting in dramatically different deep-layer geometries.

Analogy: Two trains departing from the same station with 1° directional difference. The initial gap is negligible, but after 1000km, destinations differ by tens of kilometers.

5.2 Relationship to Prior Work

Cai et al. [2024] found that LLM intermediate layers have ID peaks corresponding to critical turning points in language abstraction. Our findings are complementary:

Cai et al. (2024)	This Paper
Studies inherent layer-wise ID patterns	Studies how prompts change ID patterns
Descriptive findings	Controllable, actionable interventions
Predicts model performance	Evaluates prompt quality

5.3 Interpretability Perspective

Our findings provide new tools for LLM interpretability:

1. **Quantitative evaluation of prompt effects**: Without examining outputs, EID trajectories can indicate whether a prompt has "activated" the model's deep computational capacity.
2. **Black-box to gray-box**: While we don't know specifically what the model is "thinking," EID tells us how much computational "bandwidth" the model is using to think.
3. **Debugging tool**: If a prompt doesn't produce expected deep-layer expansion, the signal isn't strong enough or the direction is wrong.

5.4 Limitations

1. **Quantization artifacts:** Both models use quantized weights; full-precision validation is future work.
2. **First-token measurement:** We measure EID at prompt completion; generation-time dynamics remain unexplored.
3. **Correlation vs causation:** We demonstrate correlation between expert signals and EID expansion, but causal mechanisms require further research (e.g., attention pattern analysis).
4. **Downstream task validation:** We have not validated whether higher EID directly leads to better task performance.

6 Conclusion

This paper provides empirical evidence that expert-level prompts induce **Deep Layer Expansion** in LLM representations—a 60-100% increase in Effective Intrinsic Dimension at deep layers. This phenomenon holds universally across Qwen and Llama architectures, suggesting it is a fundamental geometric property of Transformer models responding to high-quality semantic signals.

We propose the **Manifold Teleportation** framework: expert signals act as navigators that guide activation trajectories away from low-dimensional collapse regions toward high-dimensional semantic manifolds. This provides a geometric foundation for understanding why prompt engineering works and offers new directions for LLM interpretability research—understanding model behavior by tracing the geometric properties of representation space.

Future Work:

1. Causal mechanism analysis: Understanding the causes of deep-layer expansion through attention patterns and circuit analysis
2. Downstream validation: Establishing quantitative relationships between EID and task performance
3. Automated tools: Automatic prompt quality evaluation based on EID
4. More models: Validation on other architectures such as Mistral and Gemma

References

Ansuini, A., Laio, A., Macke, J. H., & Zoccolan, D. (2019). Intrinsic dimension of data representations in deep neural networks. *NeurIPS*.

Cai, T., et al. (2024). Emergence of a High-Dimensional Abstraction Phase in Language Transformers. *ICLR 2024*.

Elhage, N., et al. (2022). Toy models of superposition. *Transformer Circuits Thread*.

Kirsanov, D., et al. (2025). The Geometry of Prompting: Unveiling Distinct Mechanisms of Task Adaptation in Language Models. *arXiv preprint*.

Liu, P., et al. (2023). Pre-train, prompt, and predict: A systematic survey of prompting methods in natural language processing. *ACM Computing Surveys*.

Marks, S., & Tegmark, M. (2023). The Geometry of Truth: Emergent Linear Structure in Large Language Model Representations. *NeurIPS 2023*.

Reynolds, L., & McDonell, K. (2021). Prompt programming for large language models: Beyond the few-shot paradigm. *CHI EA '21*.

Shanahan, M., McDonell, K., & Reynolds, L. (2023). Role play with large language models. *Nature*.

Valeriani, L., et al. (2023). The geometry of hidden representations of large transformer models. *NeurIPS 2023*.

Wang, X., et al. (2024). The Shape of Learning: Anisotropy and Intrinsic Dimensions in Transformer-Based Models. *EACL 2024 Findings*.

Wei, J., et al. (2022). Chain-of-thought prompting elicits reasoning in large language models. *NeurIPS*.

Zou, A., et al. (2023). Representation Engineering: A Top-Down Approach to AI Transparency. *arXiv preprint*.

A Technical Topics (50)

1. Leader Election Mechanism in Raft Consensus Algorithm
2. Frequency Domain Properties of Positional Encoding in Transformer Architecture
3. Copy-on-Write Mechanism in Operating Systems
4. Database Transaction Isolation Levels and Phantom Read Problem
5. eBPF Applications in Cloud-Native Network Observability
6. Go Language GMP Scheduling Model and Preemptive Scheduling
7. Redis Persistence Mechanisms: AOF vs RDB Trade-offs
8. Kubernetes Informer Mechanism and List-Watch

9. Differences Between JVM CMS and G1 Garbage Collectors
10. Key Exchange Algorithms in HTTPS Handshake Process
11. Kafka Zero-Copy Technology Principles
12. Security Analysis of Redlock Algorithm for Distributed Locks
13. React Fiber Architecture and Time Slicing
14. TCP Congestion Control Algorithm BBR Principles
15. B+Tree vs LSM-Tree Read/Write Performance Comparison in Storage Engines
16. Docker Container Namespace and Cgroups Isolation Principles
17. Impact of Python GIL (Global Interpreter Lock) on Multithreading
18. Multiplexing Differences Between HTTP/2 and HTTP/3 (QUIC)
19. Gradient Vanishing and Gradient Explosion Problems in Neural Networks
20. Mathematical Derivation of Bloom Filter False Positive Rate
21. Consistent Hashing Algorithm Applications in Distributed Caching
22. MySQL InnoDB MVCC Implementation Principles
23. Linux Kernel Mode and User Mode Switching Overhead
24. Git Underlying Data Structure (Merkle DAG)
25. Elasticsearch Inverted Index Compression Algorithms
26. Nginx Reverse Proxy and Load Balancing Algorithms
27. Protobuf vs JSON Serialization Performance Comparison
28. CDN Edge Caching and Origin Pull Strategies
29. OAuth 2.0 Authorization Code Flow Security
30. SYN Flood Defense Mechanisms Against DDoS Attacks
31. WebAssembly (Wasm) Sandbox Security Model
32. Rust Language Ownership and Borrow Checker
33. Non-Negotiability of P (Partition Tolerance) in CAP Theorem
34. ClickHouse Columnar Storage and Vectorized Execution
35. Prometheus Time Series Database Compression Algorithm (Gorilla)
36. Hadoop MapReduce Shuffle Process Explained
37. Differences Between Zookeeper ZAB Protocol and Paxos

38. Network Latency Analysis of Sidecar Pattern in Service Mesh
39. GraphQL vs RESTful API N+1 Problem
40. Vue.js Reactivity Principles and Dependency Collection
41. MongoDB Sharded Cluster Balancer Mechanism
42. RabbitMQ Dead Letter Queue and Delayed Message Implementation
43. Ceph Distributed Storage CRUSH Algorithm
44. Spark RDD Wide Dependency vs Narrow Dependency Division
45. Flink Backpressure Mechanism Principles
46. PostgreSQL Physical Replication vs Logical Replication
47. DNS Recursive Query vs Iterative Query Process
48. ARP Protocol Spoofing and Defense
49. CSRF Cross-Site Request Forgery Token Defense
50. SQL Injection Blind Injection Principles

B Experimental Details

B.1 Hardware Configuration

- **Platform:** DGX Spark (NVIDIA GB10, 128GB unified memory)
- **Qwen2.5-72B:** AWQ INT4 quantization (∼40GB memory)
- **Llama-3.3-70B:** W8A8 INT8 quantization (∼70GB memory)

B.2 EID Computation Code

```
import numpy as np

def compute_eid(hidden_states):
    """Compute Effective Intrinsic Dimension via spectral entropy

    Args:
        hidden_states: [batch, seq_len, hidden_dim] tensor

    Returns:
        float: Effective Intrinsic Dimension
    """
    # Take last token representation
    data = hidden_states[:, -1, :].cpu().numpy()

    # SVD decomposition
    U, S, Vh = np.linalg.svd(data, full_matrices=False)
```

```
    # Normalize singular values to probability distribution
    S_norm = S / np.sum(S)

    # Shannon entropy
    entropy = -np.sum(S_norm * np.log(S_norm + 1e-12))

    # Effective dimension = exp(entropy)
    return np.exp(entropy)
```

B.3 Reproducibility

Code and data will be released on GitHub upon paper acceptance.

C Supplementary Analysis

C.1 Inter-Topic Variance

Standard deviation across 50 topics:

- Qwen Layer 70: Standard ± 3.2, Expert ± 4.8
- Llama Layer 70: Standard ± 2.9, Expert ± 5.1

Higher variance under expert conditions reflects topic-dependent activation of specialized knowledge regions—different technical domains activate different high-dimensional subspaces.

C.2 Cross-Model Correlation

Pearson correlation coefficient between Qwen and Llama EID trajectories:

- Standard condition: $r = 0.94$
- Expert condition: $r = 0.91$

High cross-model correlation supports the universality of the observed geometric patterns—this is not a peculiarity of a single model, but a common property of Transformer architectures.