Layer by Layer: Uncovering Hidden Representations in Language Models

Oscar Skean [1] **Md Rifat Arefin** [2,3] **Dan Zhao** [4] **Niket Patel** [5] **Jalal Naghiyev** [6]
Yann LeCun [4,7] **Ravid Shwartz-Ziv** [4,8]

Presentation Outline

- Overview of Work
- Empirical Experiments on Embedding Benchmark
- Theoretical Toolkit
- Implications of our Findings

Birds Eye View of the Work

- Our work **challenges common assumptions** in modern ML folklore
    - Common Assumption : “Final-layer representations are the most useful for zero-shot downstream tasks”
    - Common Assumption : “The middle layers of an LLM are useless for token/embeddings generation”
- Our work finds that embeddings from **intermediate layers often outperform final layers** when used for downstream tasks
    - Rigorous empirical testing
    - Theoretical toolkit for analyzing internal model behavior

Massive Text Embedding Benchmark (MTEB)

- MTEB is a current state-of-the-art benchmark for evaluating LLMs on hundreds of embedding tasks

- We evaluated 32 tasks across 5 different domains for every single model layer

Task Domain	Tasks	# Tasks (32 Total)
Pair Classification	SprintDuplicateQuestions, TwitterSemEval2015, TwitterURLCorpus	3
Classification	AmazonCounterfactualClassification, AmazonReviewsClassification, Banking77Classification, EmotionClassification, MTOPDomainClassification, MTOPIntentClassification, MassiveIntentClassification, MassiveScenarioClassification, ToxicConversationsClassification, TweetSentimentExtractionClassification	10
Clustering	ArxivClusteringS2S, BiorxivClusteringS2S, MedrxivClusteringS2S, RedditClustering, StackExchangeClustering, TwentyNewsgroupsClustering	6
Reranking	AskUbuntuDupQuestions, MindSmallReranking, SciDocsRR, StackOverflowDupQuestions	4
Sentence to Sentence	BIOSSES, SICK-R, STS12, STS13, STS14, STS15, STS16, STS17, STSBenchmark	9

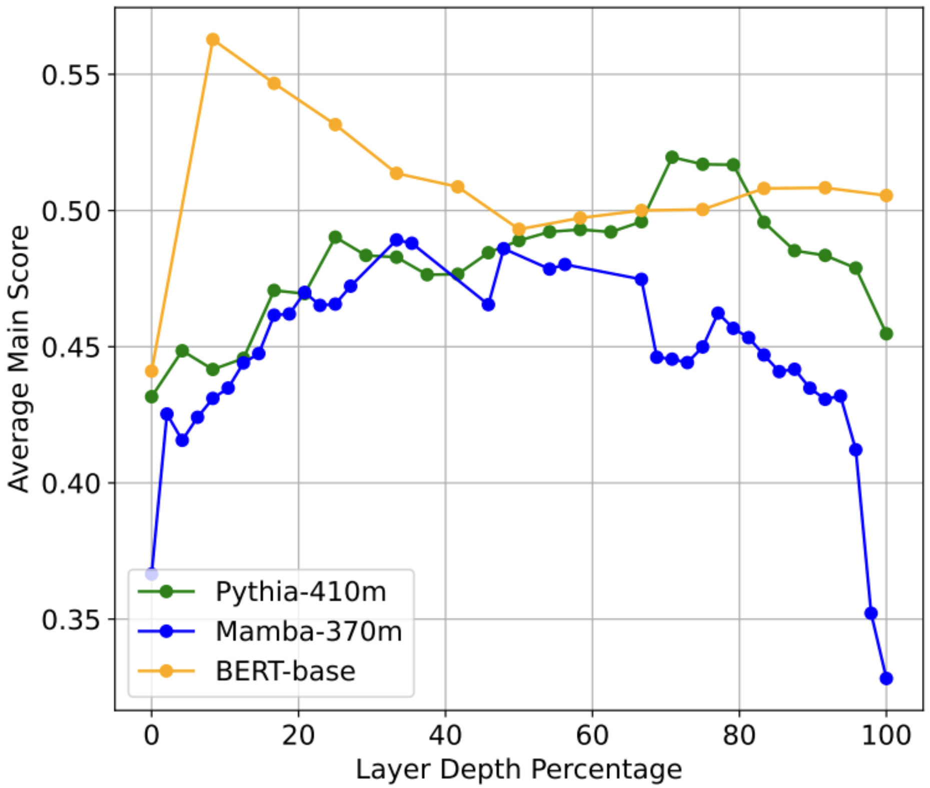


Figure 1: **Intermediate layers consistently outperform final layers on downstream tasks.** The average score of 32 MTEB tasks using the outputs of every model layer as embeddings for three different model architectures. The x-axis is the depth percentage of the layer, rather than the layer number which varies across models.

The Metrics Zoo

Geometric

- Curvature

Augmentation Invariance

- LiDAR
- DiME
- infoNCE

Information Theoretic

- Prompt Entropy
- Dataset Entropy
- Effective Rank

Proposed a framework of metrics to understand the internal model behavior

The Metrics Zoo

Geometric

- Curvature

Augmentation Invariance

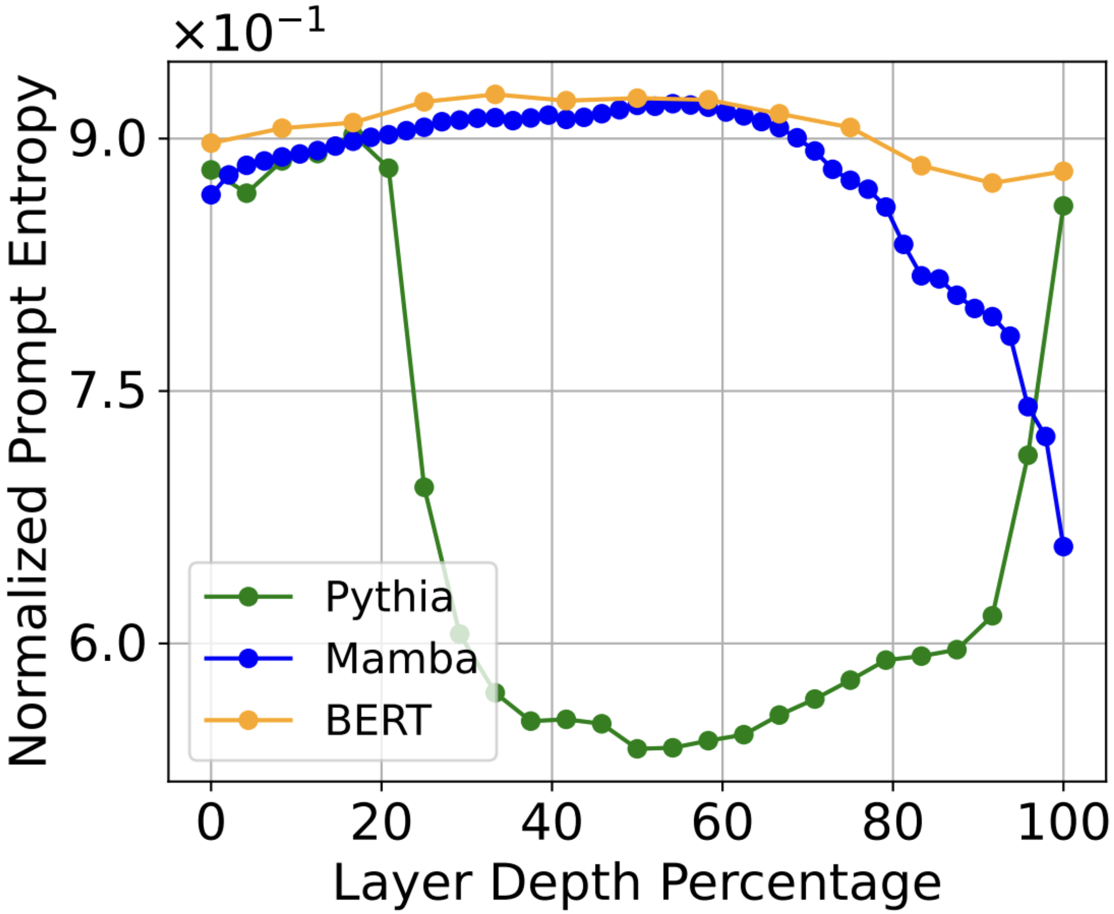
- LiDAR
- DiME (Skean et. al)
- infoNCE

Information Theoretic

- Prompt Entropy
- Dataset Entropy
- Effective Rank

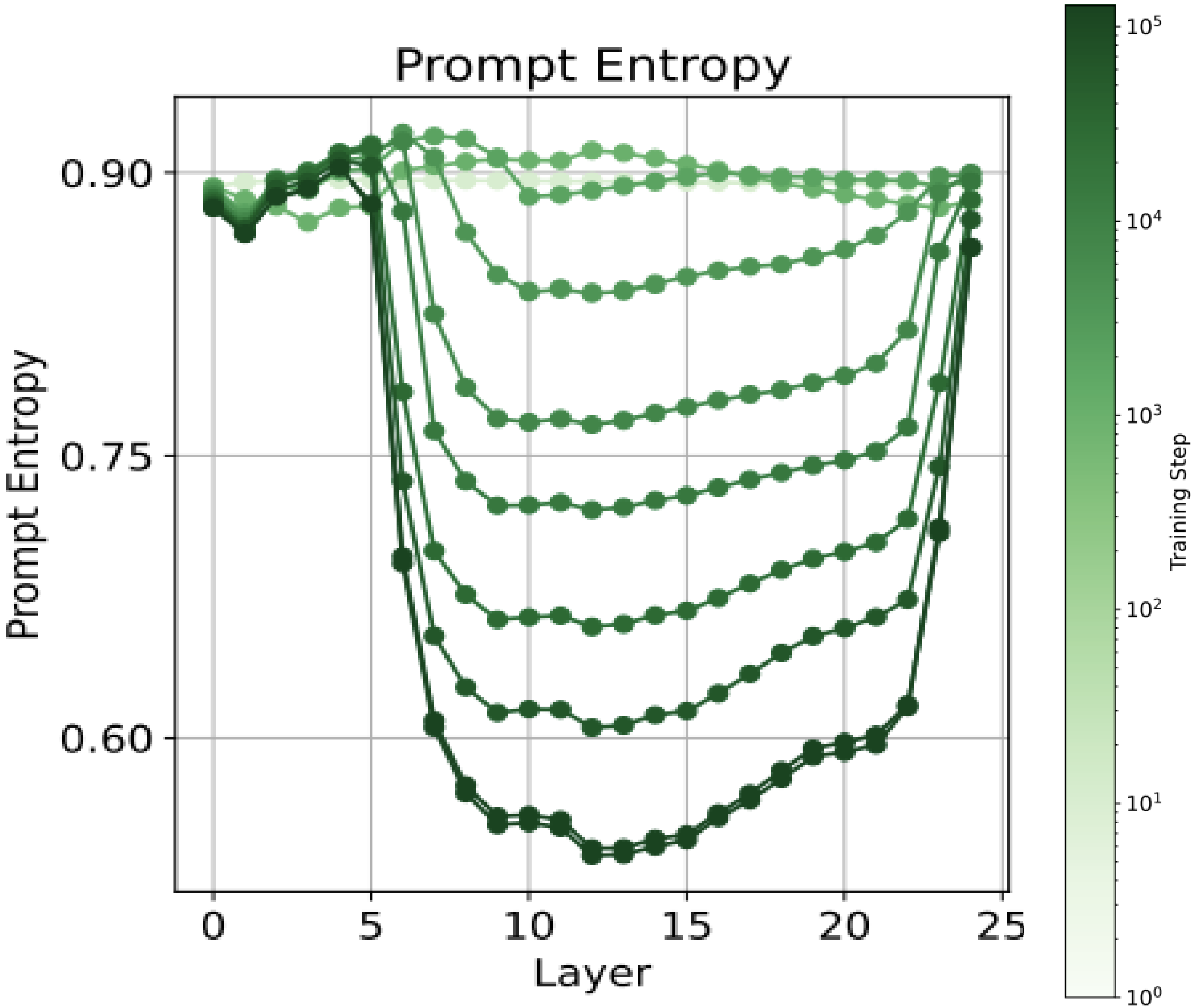
Prompt entropy captures “how compressed” representations are

Across Architectures



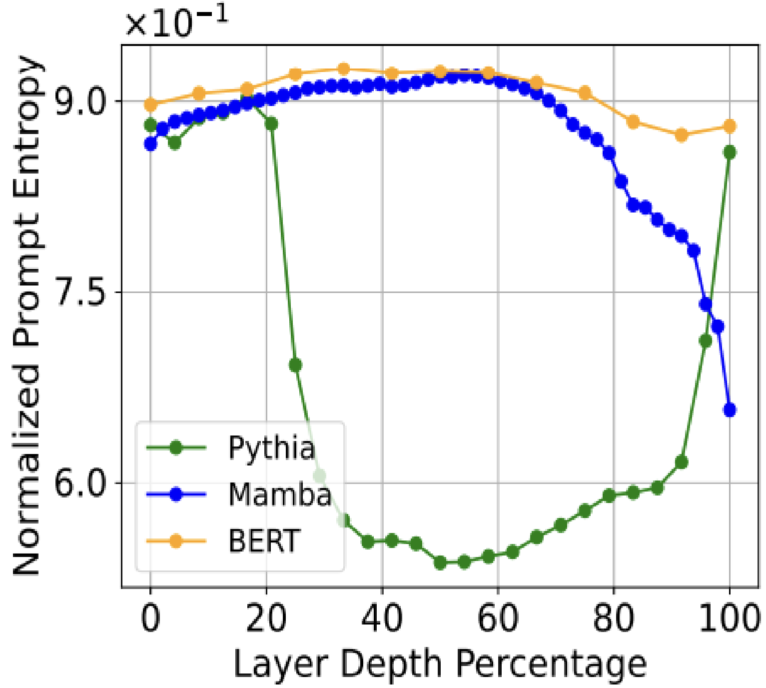
(a) Prompt Entropy

Across Training



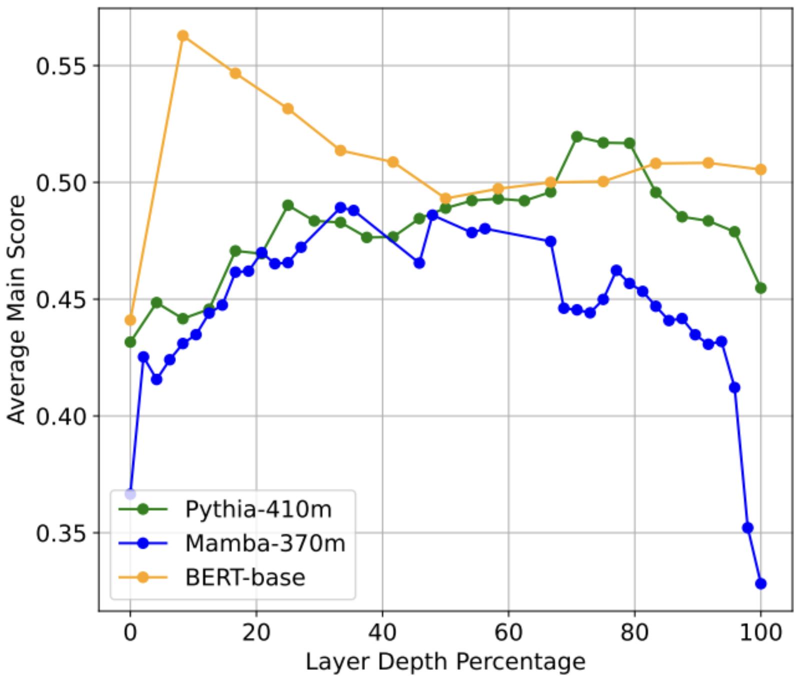
Correlations between Performance and Metrics

Entropy (unsupervised)



Correlations in autoregressive transformer models

Downstream Task Performance (Supervised)



Why It Matters

- **(Performance Boost)** Relatively easy to check if intermediate layers offer better results.

- **(Memory Footprint)** If a model has 32 layers but layer 18 is optimal, then you only need to load 18 layers into memory

- (**Understanding**) Better understanding of internal model behavior

- (**Improved Training**) Follow-up work at ICLR (Seq-VCR) used our framework to substantially improve chain-of-thought reasoning on GSM8k math tasks

Thanks!!

- Feel free to reach out to me at oscar.skean@uky.edu
- Hope to see you at our poster at ICML