

The Generalization Ridge: Information Flow in Natural Language Generation

Ruidi Chang, Chunyuan Deng, Hanjie Chen

Department of Computer Science
Rice University
Houston, TX 77005, USA
{ruidi.chang, chunyuan.deng, hanjie}@rice.edu

Abstract

Transformer-based language models have achieved state-of-the-art performance in natural language generation (NLG), yet their internal mechanisms for synthesizing task-relevant information remain insufficiently understood. While prior studies suggest that intermediate layers often yield more generalizable representations than final layers, how this generalization ability emerges and propagates across layers during training remains unclear. We propose InfoRidge, an information-theoretic framework, to characterize how predictive information—the mutual information between hidden representations and target outputs—varies across depth during training. Our experiments across various models and datasets reveal a consistent non-monotonic trend: predictive information peaks in intermediate layers—forming a **generalization ridge**—before declining in final layers, reflecting a transition between generalization and memorization. To further investigate this phenomenon, we conduct a set of complementary analyses that leverage residual scaling and attention patterns to characterize layer-wise functional specialization. We further validate our findings with multiple-token generation experiments, verifying that the observed ridge phenomenon persists across decoding steps. Together, these findings offer new insights into the internal mechanisms of transformers and underscore the critical role of intermediate layers in supporting generalization.¹

1 Introduction

Transformer-based language models generate text by predicting tokens autoregressively, and they have achieved remarkable performance across a wide range of natural-language uses (Vaswani et al., 2017; Dong et al., 2022). Nevertheless, we lack a rigorous understanding of how these models acquire and synthesize task-relevant information during training.

A growing body of research has shown that intermediate layers in deep neural networks often surpass final layers in terms of representational quality and generalization performance (Liu et al., 2019b; Voita et al., 2019; Ansuini et al., 2019; Ahrens et al., 2023; Uselis & Oh, 2025). In language models, intermediate layers often encode richer semantic and more robust features than final layers (Fan et al., 2024; Jin et al., 2024; Skean et al., 2025). However, questions still remain: *In NLG, how does information evolve across layers during training, and how are different layers of the network functionally organized to support generalization versus memorization?*

To investigate these questions, we propose *InfoRidge*, an information-theoretic framework, to analyze information flow in language models. Building on matrix-based mutual information estimation (Giraldo et al., 2014), our approach quantifies how predictive signals transform across layers during training. We center our analysis on two complementary

¹The code can be found at <https://github.com/chili-lab/InfoRidge>.

quantities: *predictive information*, defined as the mutual information $I(Z_\ell; Y)$ between the hidden representation Z_ℓ at layer ℓ and the next-token label Y , reflecting how much task-relevant information is preserved; *incremental information gain*, denoted as $I(\Delta Z_\ell; Y)$, where ΔZ_ℓ is the residual change between successive layer ℓ and $\ell - 1$, measuring the additional predictive information introduced by each transformer block.

Using *InfoRidge*, we uncover a non-monotonic trend: predictive information rises through the early and middle layers, peaks in the upper-middle layers, and then declines in the later layers. We name this phenomenon the *generalization ridge*, where the model encodes the more generalizable task-relevant information. This ridge marks a structural division of labor: intermediate layers concentrate generalizable features that transfer across distributions, while later layers increasingly specialize in task-specific memorization. Incremental information gain further shows that ridge layers introduce the largest increases in predictive information, marking them as key contributors to the emergence of the ridge. This analysis directly connects the information peak to generalizable behavior and clarifies how generalization and memorization are distributed across depth.

We further extend *InfoRidge* to multi-token natural language generation. By treating the full output sequence as a structured target and measuring mutual information between hidden states and the long-form target distribution, we show that the generalization ridge persists across decoding steps. Despite increasing mutual information as generation progresses, the layer-wise structure remains stable: predictive information consistently peaks in early-to-intermediate layers and declines in later layers.

To understand the formation of the generalization ridge, we analyze attention patterns and residual scaling coefficients. Attention analysis shows ridge layers attend to tokens that capture broadly useful features, aligning with the information peak. We also investigate the conditions under which the ridge emerges. We also introduce residual scaling coefficients β_ℓ —learnable scalars applied to each residual block with all other weights frozen—as functional probes of layer-wise reliance (Liu et al., 2019a; Menghani et al., 2024). When evaluated under distribution shift, models have lower reliance on the final layer and shift importance toward the intermediate layers.

Contributions. Our work provides a unified perspective on how predictive information is structured across depth in transformer-based language models for natural language generation tasks:

1. Our work tracks the evolution of predictive information throughout training, establishing a clear connection between predictive information flow and generalization. It reveals a non-monotonic peak in the middle layers, which we refer to as the *generalization ridge*. This pattern reflects a meaningful transition in representational focus and aligns with stronger generalization behavior.
2. We introduce *InfoRidge*, an information-theoretic framework that applies matrix-based mutual information estimation to autoregressive language models to analyze information flow.
3. We analyze attention patterns and residual scaling coefficients, providing causal evidence for the generalization ridge phenomenon.

2 InfoRidge: information estimation framework

We propose *InfoRidge*, an information-theoretic framework that uses mutual information to quantify how predictive information propagates through transformer layers during training in NLG.

Motivating Insight. Prior work has shown that internal representations in deep neural networks tend to align most closely with the true data distribution at an intermediate layer (He et al., 2024). By employing the Wasserstein distance (Villani et al., 2008), this alignment is shown to reach a minimum at a specific depth—referred to as the *generalization funnel layer*. At this point, the *Min Wasserstein Generalization Bound* (He et al., 2024) ensures

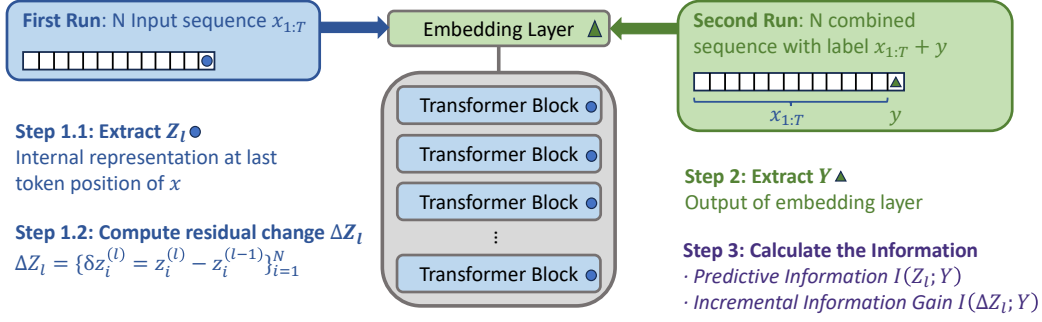


Figure 1: Overview of InfoRidge. (1) Step 1: Extract internal representations at each layer and compute residual changes between successive layers. (2) Step 2: Extract the target token embedding. (3) Step 3: Using these representations, we estimate the layer-wise predictive information $I(Z_\ell; Y)$ and the incremental information gain $I(\Delta Z_\ell; Y)$, which jointly characterize how information evolves across depth.

that the upper bound on the generalization gap—defined as the expected difference between the population and empirical risks—is minimized. This highlights the critical role of intermediate layers in supporting generalization.

Research Question. Despite the insight from prior studies of deep neural networks in classification, it remains unclear whether this generalizes to *Transformer-based language models in natural language generation (NLG)*. This motivates our central question: **In NLG, how does information evolve across layers during training, and how are different layers of the network functionally organized to support generalization versus memorization?**

Hypothesis. Building on the insight, we hypothesize that there exists a specific intermediate transformer layer that encodes the most generalizable representations for next-token prediction, characterized by maximal mutual information with the target label. We refer to this layer as the *generalization ridge*.

Hypothesis: Generalization Ridge

There exists an intermediate layer $\ell^* \in \{1, \dots, L\}$ such that the mutual information between the hidden state and the target label peaks at that layer:

$$\ell^* = \arg \max_{\ell} I(Z_\ell; Y).$$

We refer to this layer as the *generalization ridge*. This ridge layer aligns most strongly with generalizable features and serves as robust predictors under distribution shift, whereas later layers increasingly specialize in memorization.

Notation and Setup. We consider a transformer model with L residual blocks. Given an input sequence $x_{1:T}$ of length T , $z_i^{(\ell)} \in \mathbb{R}^d$ denotes the hidden state at the last token position of the i th input sequence in layer ℓ , for $\ell = 1, \dots, L$. For next-token prediction, the ground-truth label is denoted by $y_i \in \mathbb{R}^d$, corresponding to the embedding of the true next token from the vocabulary $\mathcal{V}$. The residual transformation introduced at layer ℓ is defined as $\delta z_i^{(\ell)} = z_i^{(\ell)} - z_i^{(\ell-1)}$.

Across a batch of N sequences, we collect the representations:

$$Z_\ell = \{z_i^{(\ell)}\}_{i=1}^N, \quad Y = \{y_i\}_{i=1}^N, \quad \Delta Z_\ell = \{\delta z_i^{(\ell)}\}_{i=1}^N,$$

where all vectors are ℓ_2 -normalized.

InfoRidge Overview. To empirically investigate this hypothesis, we propose InfoRidge, an information estimation framework, that characterizes how predictive information evolves across transformer layers. Specifically, we estimate two key quantities:

- **Predictive Information** $I(Z_\ell; Y)$: the mutual information between the hidden state at layer ℓ and the target token. This quantity measures how much information about the true next token is contained in the layer’s full representation. A high value indicates that the layer encodes a strong and direct signal relevant to the prediction task.
- **Incremental Information Gain** $I(\Delta Z_\ell; Y)$: the information introduced by the residual transformation at layer ℓ , where $\Delta Z_\ell = Z_\ell - Z_{\ell-1}$. This captures the additional predictive signal gained through the residual transformation at layer ℓ , isolating how much new task-relevant information is introduced on top of the previous layer’s representation.

Together, these metrics allow us to track both the accumulation and transformation of task-relevant information throughout the network.

Computational Flow Overview. Figure 1 illustrates the workflow used to extract intermediate representations for information analysis. In the first forward pass, a batch of N input sequences $x_{1:T}$ is fed into the transformer to obtain hidden states at each layer. From these, we extract the final-token representations Z_ℓ and compute the residual changes ΔZ_ℓ by differencing consecutive layer outputs. In the second forward pass, each input is concatenated with its ground-truth next token y , and we extract the corresponding label embedding Y from the output of the embedding layer. These representations are then used to compute two information-theoretic quantities: the *Predictive Information* $I(Z_\ell; Y)$, and the *Incremental Information Gain* $I(\Delta Z_\ell; Y)$. We estimate both $I(Z_\ell; Y)$ and $I(\Delta Z_\ell; Y)$ using Equations 1 and 2, detailed below.

Matrix-Based Mutual Information. We apply the matrix-based framework (Giraldo et al., 2014) to estimate mutual information. Let $\mathcal{U}$ be a random variable, from which we draw a set of vectors $U = \{\mathbf{u}_i\}_{i=1}^N \subset \mathbb{R}^d$. A positive-definite Gram matrix $G_U \in \mathbb{R}^{N \times N}$ is computed using a Gaussian kernel κ with bandwidth set to 1 and the matrix is then trace-normalized to satisfy $\text{tr}(G_U) = 1$. To assess robustness with respect to kernel selection, we additionally test the Laplacian and Polynomial kernels. While these kernels introduce quantitative shifts in the magnitude, the trend remains unchanged, see Appendix E. The matrix-based Rényi entropy (with order $\alpha = 1$) is then given by:

$$H(\mathcal{U}) \approx H(U) = -\text{tr}(G_U \log G_U). \quad (1)$$

Specifically, G_U is constructed with entries $(G_U)_{ij} = \exp\left(-\frac{\|\mathbf{u}_i - \mathbf{u}_j\|^2}{2\sigma^2}\right)$ and then trace-normalized.

Let G_U and G_V be the trace-normalized Gram matrices for two random variables $\mathcal{U}$ and $\mathcal{V}$, respectively. The mutual information between them is computed as:

$$\begin{aligned} I(\mathcal{U}; \mathcal{V}) &\approx I(G_U; G_V) \\ &= H(G_U) + H(G_V) - H(G_U \circ G_V), \end{aligned} \quad (2)$$

where “ $\circ$ ” denotes the Hadamard product. Mathematical details are in Appendix A.

3 Experimental setup

Models. We evaluate four models: GPT-2 SMALL (117M) (Radford et al., 2019), GPT-2 MEDIUM (345M) (Radford et al., 2019), QWEN-2.5 0.5B (Yang et al., 2024), and LLAMA 3.1 8B (Meta AI, 2024). For the experiments in Section 4, all models are fine-tuned on their respective downstream tasks.

Datasets. We assess model behavior across four tasks casted into NLG problems: CLUTRR (Sinha et al., 2019), a relational reasoning benchmark; ECQA (Aggarwal et al., 2021), a commonsense QA benchmark; CNN/DailyMail (See et al., 2017; Hermann et al., 2015), a summarization benchmark and Synthetic Arithmetic, a controlled dataset designed to disentangle task-relevant signal from noise. Dataset and implementation details are in Appendix B and C.

Synthetic Arithmetic Dataset Construction. We construct a synthetic dataset to separate signal learning from noise memorization. Each example is a sequence of 10 elements, where the signal follows an arithmetic progression modulo K , computed as $s_t = (s_0 + t \cdot d) \bmod K$ with $s_0 \in [0, K-1]$ and $d \in [1, K-1]$. Each element takes the form $S\{\text{signal}\}_N\{\text{noise}\}$, where noise is sampled from $\mathcal{U}_{\text{int}}(0, \text{noise_range}-1)$ ($\mathcal{U}_{\text{int}}$ denotes the uniform distribution). The model is trained to predict the signal value of the final (10th) element using the preceding elements as input context. For example, with $K = 5$, $s_0 = 1$, and $d = 2$, a sample input might be $S1_N42 \ S3_N77 \dots S2_N37$, with the target signal being 4. By varying K , we can induce structured distribution shifts.

4 Generalization Ridge: layer-wise mutual information training dynamics

Understanding how layers encode task-relevant information is key to uncovering the internal mechanisms that support generalization in language models. In this section, we trace the evolution of two complementary forms of mutual information—*Predictive Information* and *Incremental Information Gain*—across transformer depth and training time, revealing a structure in information flow and highlighting the generalization–memorization trade-off.

4.1 Predictive Information: information peaks at intermediate layers

We investigate how *predictive information*—defined as the mutual information between hidden representations and target labels—evolves across the depth of transformer models. Specifically, for each layer ℓ , we compute the matrix-based mutual information $I(Z_\ell; Y)$ between the hidden state Z_ℓ and the next-token ground truth Y . This quantity measures how much task-relevant signal is retained in the representation as it propagates through the network. By tracing $I(Z_\ell; Y)$ across layers, we obtain a layer-wise trajectory of information flow, which reveals not only where predictive content is preserved but also how it is transformed or diminished as the model processes.

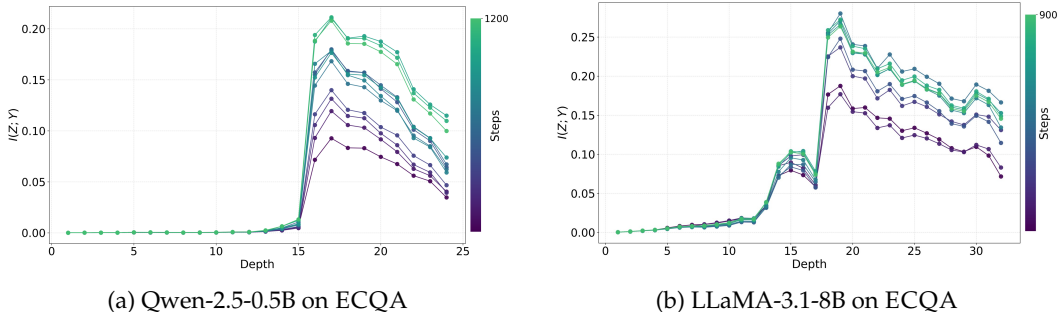


Figure 2: Evolution of predictive information $I(Z_\ell; Y)$, with lighter curves indicating later training steps. Each curve exhibits a three-phase trend: early layers rise, mid layers peak, and late layers decline.

Figure 2 tracks the trajectory of the predictive information $I(Z_\ell; Y)$ between the hidden representation at depth ℓ and the target label Y throughout training, while Table 1 reports

the downstream accuracy obtained when we early exit after a given layer.² Additional results are in Appendix E.

Three-phase Information Dynamics. Across models and tasks, predictive information curves exhibit a consistent *three-phase* pattern:

Progressive Accrual (early layers). In the initial layers, $I(Z_\ell; Y)$ gradually increases, corresponding to basic feature extraction without substantial task-level comprehension. This aligns with the near-zero accuracy observed in Table 1 for these layers.

Information Peak (intermediate layers). $I(Z_\ell; Y)$ continues to rise through the mid-to-upper layer, typically peaking before the final few blocks. For the GPT-2 Small on Synthetic Arithmetic dataset, the peak reaches $I \approx 0.27$ in the 10th layer, coinciding with a jump from $\approx 0\%$ to 57% OOD accuracy. These layers appear to play a critical role in synthesizing abstract features that are essential for generalization.

Representational Compression (final layers). Beyond the peak, $I(Z_\ell; Y)$ decreases, even as in-distribution accuracy approaches 93%. The simultaneous drop in OOD accuracy indicates that the final layers tend to memorize training patterns, sacrificing generalization ability.

Generalization Ridge: Generalization-Memorization Trade-off. The pronounced “information funnel” around intermediate layers reflects a key trade-off between generalization and memorization, which we term the “**generalization ridge**”. These layers maximize task-relevant information for generalization, while deeper layers increasingly compress and specialize representations, enhancing in-distribution memorization but reducing robustness. This positions intermediate layers as critical control points for managing this trade-off.

Layer	$I(Z; Y)$	Test Accuracy (%)		
		All	In-Dist.	Out-Dist.
Layer 1	0.0002	0.00	0.00	0.00
Layer 2	0.0003	0.00	0.00	0.00
Layer 3	0.0005	0.00	0.00	0.00
Layer 4	0.0013	0.00	0.00	0.00
Layer 5	0.0043	0.00	0.00	0.00
Layer 6	0.0185	0.00	0.00	0.00
Layer 7	0.0416	2.67	0.20	5.13
Layer 8	0.1883	17.78	0.20	35.37
Layer 9	0.2301	20.90	1.47	40.33
Layer 10	0.2721	55.15	53.40	56.90
Layer 11	0.2475	65.92	79.17	52.67
Layer 12	0.0188	71.58	93.17	50.00

Table 1: Information dynamics and layer-wise performance (GPT-2-S, Syn). OOD performance declines beyond the ridge.

Semantically Important Attention Peaks Where Information Peaks. This discussion aims to provide an intuitive illustration of our generalization ridge hypothesis, highlighting how attention to task-relevant signal tokens shifts across layers (the generalizable information). Specifically, we computed average signal attention across layers, identified the layer with peak signal attention, and compared it to final-layer signal and last-token attention, results are in Table 2. Signal tokens are defined as task-relevant tokens that the model must focus on to solve the task—for the synthetic dataset, these are tokens that appear after the character ‘S’; for CLUTRR, kinship-related entities; and for ECQA, the token corresponding to the correct answer option. This allows us to quantify where in the network attention to semantically important information is concentrated. Our findings show a pattern that supports our hypothesis: (1) Mid-to-late layers peak in signal attention, coinciding with the predictive information ridge (Figure 2), indicating where generalizable representations are strongest. (2) Final layers show reduced attention to signal tokens, suggesting a shift toward memorization to specific data points rather than predictive information abstraction. For example, in Qwen-2.5-0.5B on ECQA, signal attention peaks at Layer 17 (0.2307) but drops to 0.0104 in the final layer, where last-token attention dominates (0.2169).

4.2 Incremental Information Gain: information concentrates at intermediate layers

²The in distribution split is generated with $K_{id}=13$; out of distribution splits use a uniformly-sampled $K_{ood} \in \{5, \dots, 25\} \setminus \{13\}$.

Model	Dataset	Avg. Attn (All)	Avg. Attn (Signal)	Layer w/ Highest Avg. Signal	Final Avg. Signal	Final Avg. Last
Qwen-2.5-0.5B	ECQA	0.0229	0.0317	17 (0.2307)	0.0104	0.2162
LLaMA-3.1-8B	ECQA	0.0224	0.0394	21 (0.1595)	0.0094	0.5073
GPT-2 Small	Synthetic	0.0222	0.0409	8 (0.0739)	0.0555	0.0378
GPT-2 Medium	CLUTRR	0.0080	0.0078	18 (0.0243)	0.0112	0.1576

*We remove the first token attention score to mitigate attention sink effects.

Table 2: Average attention statistics: (1) average attention scores over all tokens, (2) average attention to signal tokens, (3) the maximum signal attention and its corresponding layer, (4) signal token attention in the final layer and (5) last token attention in the final layer.

To understand how information accumulates across the network, we compute *Incremental Information Gain* ($I(\Delta Z_\ell; Y)$)—the mutual information between each residual transition and the target label embedding. As shown in Figure 3, the resulting layer-wise gains reveal that intermediate layers yield the highest information increases. This concentration of information gain further underscores their central role in encoding those task-relevant features that are essential for supporting generalization. The Incremental Information Gain analysis reveals a clear pattern: middle transformer blocks are key to encoding generalizable task-relevant information, thereby forming a *generalization ridge*. Conversely, later layers contribute little additional predictive signal, and in some cases, actively reduce alignment with the target embeddings. This diminishing contribution in later layers may suggest a shift from general reasoning to memorization of training-specific patterns. This trend underscores a fundamental trade-off in transformer training dynamics. Additional results are in Appendix E.

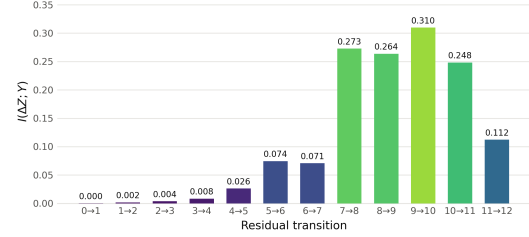


Figure 3: Middle blocks are key to encoding generalizable information (GPT-2-S, Synthetic).

Residual Scaling Dynamics via Learnable β Coefficients. To deepen our understanding of the Generalization Ridge hypothesis, we examine how modulating the contribution of individual transformer blocks affects information flow.

We introduce a *residual scaling mechanism* with learnable scalar coefficient parameters, inspired by prior work on adaptive residual modulation (Liu et al., 2019a; Menghani et al., 2024). Transformer architectures inherently employ residual connections to iteratively refine representations. To isolate and quantify the contribution of each transformer block, we scale these residual connections with layer-specific scaling factors $\beta_\ell \in \mathbb{R}_{\geq 0}$, with definition below. Each β_ℓ controls the strength of the residual contribution from layer ℓ , enabling the model to adaptively emphasize or suppress specific blocks:

$$z^{(\ell)} = z^{(\ell-1)} + \beta_\ell \cdot \text{block}^{(\ell)}(z^{(\ell-1)}), \quad \beta_\ell \in \mathbb{R}.$$

We freeze model weights and optimize only the residual scaling coefficient parameters β_ℓ , which are initialized to 1. These scalars are trained separately on the (a) in-distribution (ID) split and (b) out-of-distribution (OOD) split. Since no other parameters are updated, the learned β_ℓ serve as a direct diagnostic of the extent to which each layer should be amplified or attenuated to suit the data regime, revealing which layers remain stable across regimes and which adapt strongly to distributional shifts.

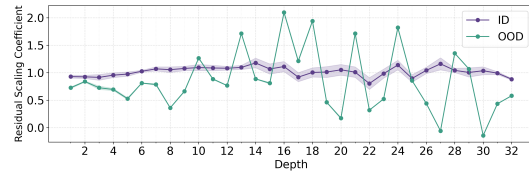


Figure 4: Residual scaling coefficients β_ℓ across all layers. Curve shows the mean across 5 independent runs, and the shaded region denotes 1- σ error bar. LLaMA on ECQA: training on OOD data improves OOD performance from 0.0% to 21.1%.

As shown in Figure 4, optimizing the residual scaling coefficient parameters on in-distribution data assigns relatively higher residual weights to the final transformer layers, indicating that in-distribution performance benefits from increased reliance on late-layer residual contributions. This observation suggests that these deeper layers capture refinements that are closely tied to the training distribution. In contrast, under out-of-distribution (OOD) settings, the learned coefficients systematically have relative lower contributions of the final layers compared to the ID settings, revealing that improved generalization is associated with downweighting late-stage residual signals. When train on OOD, intermediate layers are upweighted and OOD performance improves. Additional results are in Appendix E. This pattern supports the interpretation that late-layer residuals become increasingly specialized in memorized patterns tied to the training distribution. Together, these findings provide empirical support for the generalization ridge hypothesis, demonstrating that information flow in transformers reflects a trade-off between generalizable signals and memorized, distribution-specific features.

5 Extension to multi-token scenario

We further evaluate information flow under multi-token outputs using the CNN/DailyMail summarization dataset (1.0.0, test) (See et al., 2017; Hermann et al., 2015). We evaluate two pretrained models: Qwen-2.5-0.5B and LLaMA-3.1-8B, generating 50 tokens per sample.

Kernel Extension for Variable-Length Sequences. To handle multi-token targets, we extend our matrix-based mutual information (MI) estimation to variable-length sequences. Let $Y_i = \{y_i^{(t)}\}_{t=1}^{T_i}$ denote the target sequence for sample i . We employ the following kernel to compute sample-to-sample similarity:

$$K(Y_i, Y_j) = \frac{1}{T_i T_j} \sum_{t=1}^{T_i} \sum_{t'=1}^{T_j} \kappa(y_i^{(t)}, y_j^{(t')}),$$

where $\kappa(\cdot, \cdot)$ is a Gaussian kernel. This formulation aggregates all pairwise token similarities between two target sequences into a single scalar, capturing the overall alignment between two token distributions. We quantify how much information each layer

carries about the gold answer by computing $I(Z_\ell^{(t)}; Y)$ in Figure 5, where Z_ℓ denotes the hidden state at the last token position at decoding step t and Y represents the full target sequence.

Mutual information across generation steps. We observe several consistent patterns in the mutual information profiles across generation steps. First, the overall magnitude of $I(Z_\ell^{(t)}; Y)$ increases as generation progresses: later decoding steps tend to exhibit higher mutual information than earlier ones. This reflects the accumulation of information in the hidden representations as the model conditions on an increasingly informative prefix composed of its previously generated tokens. As generation progresses, the hidden state progressively approach the target sequence distribution. Second, despite differences in magnitude, the layer-wise structure of the mutual information curves remains consistent across generation steps. In particular, $I(Z_\ell^{(t)}; Y)$ peaks at early-to-intermediate layers and gradually declines toward later layers. The concentration of mutual information in early-to-intermediate layers reflects their role in forming task-relevant representations under autoregressive decoding. These layers integrate information from the input and the evolving context into relatively abstract features that are shared across decoding steps, such as semantic content and task-level structure. While these representations are critical for producing the next output token,

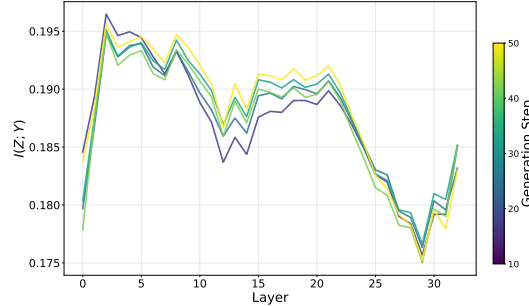


Figure 5: $I(Z_\ell^{(t)}; Y)$ across layers (x-axis) and generation steps (right color bar), the y-axis shows mutual information values. Values are plotted every ten steps.

later layers increasingly compress information into a lower-entropy, leading to reduced mutual information with the target sequence.

6 Related work

Understanding how information is encoded and transformed across layers has been studied through probing classifiers (Alain & Bengio, 2016), attention flow (Vig & Belinkov, 2019), and information-theoretic approaches such as the information bottleneck (Shwartz-Ziv & Tishby, 2017), mutual information estimation (Goldfeld, 2019), and matrix-based entropy (Giraldo et al., 2014), offering different lenses to quantify representational capacity, abstraction, and invariance across layers.

A growing body of research has shown that intermediate layers in deep networks often outperform final layers in terms of representational quality and task performance (Ansuini et al., 2019; Yosinski et al., 2014; Uselis & Oh, 2025; Ahrens et al., 2023; Ando et al., 2023). In language models, mid-depth layers tend to capture richer semantic or robust features than output layers (Liu et al., 2019b; Voita et al., 2019; Jin et al., 2024; Fan et al., 2024). Transformer representations have long been observed to follow a structured progression from syntactic to semantic information, as shown by classical probing studies on linguistic knowledge and the reconstruction of the NLP pipeline (Liu et al., 2019b; Tenney et al., 2019). These findings challenge the assumption that deeper layers always yield better representations. More recent work further strengthens this observation by explicitly characterizing the role of middle-to-upper layers. Queipo-de Llano et al. (2026) connect the emergence of attention sinks to representational compression in intermediate layers and propose a “Mix–Compress–Refine” view of information flow across model depth. The semantic hub hypothesis suggests that these layers act as convergence points for abstract semantic representations, while other studies identify an abstraction phase in deeper layers (Wu et al., 2024; Cheng et al., 2024). Earlier probing analyses similarly reveal a structured progression of representations across layers (Tenney et al., 2019; Sajjad et al., 2022), and layer pruning studies further highlight their non-uniform functional roles (Sajjad et al., 2023). Earlier probing analyses similarly reveal a structured progression of representations across layers (Tenney et al., 2019; Sajjad et al., 2022), and layer pruning studies further highlight their non-uniform functional roles (Sajjad et al., 2023). Complementing these analyses, Ng (2026) shows that selectively repeating a contiguous block of intermediate transformer layers at inference time can improve model performance without modifying the model weights, providing additional intervention-based evidence that layers differ substantially in their functional contributions.

This pattern holds across settings such as transfer learning (Yosinski et al., 2014), continual learning (Ahrens et al., 2023), and out-of-distribution generalization (Uselis & Oh, 2025). Furthermore, recent work has evaluated representation quality using entropy, curvature, and invariance (Skean et al., 2025), while other studies have examined embedding drift and representational geometry (Merchant et al., 2020; Dar et al., 2022), analyzed memorization and factual recall (Haviv et al., 2022; Yu et al., 2023), and introduced causal perspectives on layer importance through mediation analysis and targeted interventions (Vig et al., 2020; Meng et al., 2022). Training-dynamics studies investigated how earlier models develop and refine semantic features across depth, providing an additional perspective on the evolution of layer-wise representations (Merchant et al., 2020; Kumar et al., 2023). Correlational probes (e.g., linear probes (Alain & Bengio, 2016)) measure only whether a feature can be decoded from a representation, which reflects correlation but not causal influence. In contrast, causal methods intervene on internal activations to test how changing a component alters the model’s prediction, thereby identifying true causal contribution rather than mere feature presence. Our residual-scaling approach aligns with this causal perspective at layer level.

However, the underlying causes and functional role of this phenomenon remain only partially understood, motivating further investigation. Our work addresses this gap by tracing the evolution of predictive information throughout training and establishing a clear connection between predictive information flow and generalization. We reveal a consistent non-monotonic peak in the middle layers—termed the *generalization ridge*—that reflects

a meaningful transition in representational focus and aligns with stronger generalization behavior. Additionally, unlike prior work focused on classification tasks, we extend the analysis of information flow to generation tasks, enabling us to understand how generalization and memorization dynamics evolve for natural language generation setting from an information-theoretic perspective.

7 Conclusion

We introduce InfoRidge, an information-theoretic framework designed to trace and quantify how information evolves across layers in transformer-based language models for natural language generation. By estimating both predictive information and incremental information gain, we systematically characterize the layerwise dynamics of information flow, offering a principled view of how signals are refined, amplified, or diminished as they propagate through the network. Our findings reveal a consistent *generalization ridge* emerging in intermediate layers, where mutual information between the hidden representation and the target label reaches its peak before gradually declining. This phenomenon reflects a fundamental trade-off between generalization and memorization as information flows deeper into the model. A similar pattern is observed in the multi-token setting. Attention analysis and residual scaling experiments offer complementary causal evidence supporting this observation. Taken together, these findings position InfoRidge as a comprehensive framework for diagnosing how language models internally manage information during natural language generation, while also revealing the structural mechanisms that govern the balance between generalization and memorization.

Acknowledgments

This project was supported by the Rice Ken Kennedy Institute Research Award No. 1081025 and the U.S. National Institutes of Health under Award Number OT2OD038051. The content is solely the responsibility of the authors and does not necessarily represent the official views of the National Institutes of Health.

References

- Shourya Aggarwal, Divyanshu Mandowara, Vishwajeet Agrawal, Dinesh Khandelwal, Parag Singla, and Dinesh Garg. Explanations for commonsenseqa: New dataset and models. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pp. 3050–3065, 2021.
- Kyra Ahrens, Hans Hergen Lehmann, Jae Hee Lee, and Stefan Wermter. Read between the layers: Leveraging multi-layer representations for rehearsal-free continual learning with pre-trained models. *arXiv preprint arXiv:2312.08888*, 2023.
- Guillaume Alain and Yoshua Bengio. Understanding intermediate layers using linear classifier probes. *arXiv preprint arXiv:1610.01644*, 2016.
- Atsushi Ando, Ryo Masumura, Akihiko Takashima, Satoshi Suzuki, Naoki Makishima, Keita Suzuki, Takafumi Moriya, Takanori Ashihara, and Hiroshi Sato. On the use of modality-specific large-scale pre-trained encoders for multimodal sentiment analysis. In *2022 IEEE Spoken Language Technology Workshop (SLT)*, pp. 739–746. IEEE, 2023.
- Alessio Ansuini, Alessandro Laio, Jakob H Macke, and Davide Zoccolan. Intrinsic dimension of data representations in deep neural networks. *Advances in Neural Information Processing Systems*, 32, 2019.
- Emily Cheng, Diego Doimo, Corentin Kervadec, Iuri Macocco, Jade Yu, Alessandro Laio, and Marco Baroni. Emergence of a high-dimensional abstraction phase in language transformers. *arXiv preprint arXiv:2405.15471*, 2024.

- Guy Dar, Mor Geva, Ankit Gupta, and Jonathan Berant. Analyzing transformers in embedding space. *arXiv preprint arXiv:2209.02535*, 2022.
- Chenhe Dong, Yinghui Li, Haifan Gong, Miaoxin Chen, Junxin Li, Ying Shen, and Min Yang. A survey of natural language generation. *ACM Computing Surveys*, 55(8):1–38, 2022.
- Siqi Fan, Xin Jiang, Xiang Li, Xuying Meng, Peng Han, Shuo Shang, Aixin Sun, Yequan Wang, and Zhongyuan Wang. Not all layers of llms are necessary during inference. *arXiv preprint arXiv:2403.02181*, 2024.
- Luis Gonzalo Sanchez Giraldo, Murali Rao, and Jose C Principe. Measures of entropy from data using infinitely divisible kernels. *IEEE Transactions on Information Theory*, 61(1): 535–548, 2014.
- Ziv Goldfeld. Estimating information flow in deep neural networks. In *International Conference on Machine Learning*, 2019.
- Adi Haviv, Ido Cohen, Jacob Gidron, Roei Schuster, Yoav Goldberg, and Mor Geva. Understanding transformer memorization recall through idioms. *arXiv preprint arXiv:2210.03588*, 2022.
- Haiyun He, Christina Lee Yu, and Ziv Goldfeld. Information-theoretic generalization bounds for deep neural networks. *arXiv preprint arXiv:2404.03176*, 2024.
- Karl Moritz Hermann, Tomas Kocisky, Edward Grefenstette, Lasse Espeholt, Will Kay, Mustafa Suleyman, and Phil Blunsom. Teaching machines to read and comprehend. In C. Cortes, N. Lawrence, D. Lee, M. Sugiyama, and R. Garnett (eds.), *Advances in Neural Information Processing Systems*, volume 28. Curran Associates, Inc., 2015. URL https://proceedings.neurips.cc/paper_files/paper/2015/file/afdec7005cc9f14302cd0474fd0f3c96-Paper.pdf.
- Mingyu Jin, Qinkai Yu, Jingyuan Huang, Qingcheng Zeng, Zhenting Wang, Wenyue Hua, Haiyan Zhao, Kai Mei, Yanda Meng, Kaize Ding, et al. Exploring concept depth: How large language models acquire knowledge and concept at different layers? *arXiv preprint arXiv:2404.07066*, 2024.
- Tanishq Kumar, Blake Bordelon, Samuel J Gershman, and Cengiz Pehlevan. Grokking as the transition from lazy to rich training dynamics. In *The twelfth international conference on learning representations*, 2023.
- Fenglin Liu, Meng Gao, Yuanxin Liu, and Kai Lei. Self-adaptive scaling for learnable residual structure. In *Proceedings of the 23rd Conference on Computational Natural Language Learning (CoNLL)*, pp. 862–870, 2019a.
- Nelson F Liu, Matt Gardner, Yonatan Belinkov, Matthew E Peters, and Noah A Smith. Linguistic knowledge and transferability of contextual representations. In *Proceedings of NAACL-HLT*, pp. 1073–1094, 2019b.
- Kevin Meng, David Bau, Alex Andonian, and Yonatan Belinkov. Locating and editing factual associations in gpt. *Advances in neural information processing systems*, 35:17359–17372, 2022.
- Gaurav Menghani, Ravi Kumar, and Sanjiv Kumar. Laurel: Learned augmented residual layer. *arXiv preprint arXiv:2411.07501*, 2024.
- Amil Merchant, Elahe Rahimtoroghi, Ellie Pavlick, and Ian Tenney. What happens to bert embeddings during fine-tuning? *arXiv preprint arXiv:2004.14448*, 2020.
- Meta AI. Meta llama 3.1 8b. <https://huggingface.co/meta-llama/Llama-3.1-8B>, 2024. Accessed: 2025-05-12.
- David Noel Ng. Llm neuroanatomy: How i topped the llm leaderboard without changing a single weight. March 2026. URL <https://dnhkng.github.io/posts/rys/>.

- Enrique Queipo-de Llano, Álvaro Arroyo, Federico Barbero, Xiaowen Dong, Michael Bronstein, Yann LeCun, and Ravid Shwartz-Ziv. Attention sinks and compression valleys in llms are two sides of the same coin. In *International Conference on Learning Representations*, volume 2026, pp. 13911–13937, 2026.
- Alec Radford, Jeff Wu, Rewon Child, David Luan, Dario Amodei, and Ilya Sutskever. Language models are unsupervised multitask learners. 2019.
- Hassan Sajjad, Nadir Durrani, Fahim Dalvi, Firoj Alam, Abdul Khan, and Jia Xu. Analyzing encoded concepts in transformer language models. In *Proceedings of the 2022 Conference of the North American chapter of the Association for Computational Linguistics: Human language technologies*, pp. 3082–3101, 2022.
- Hassan Sajjad, Fahim Dalvi, Nadir Durrani, and Preslav Nakov. On the effect of dropping layers of pre-trained transformer models. *Computer Speech & Language*, 77:101429, 2023.
- Abigail See, Peter J. Liu, and Christopher D. Manning. Get to the point: Summarization with pointer-generator networks. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 1073–1083, Vancouver, Canada, July 2017. Association for Computational Linguistics. doi: 10.18653/v1/P17-1099. URL <https://www.aclweb.org/anthology/P17-1099>.
- Ravid Shwartz-Ziv and Naftali Tishby. Opening the black box of deep neural networks via information. *arXiv preprint arXiv:1703.00810*, 2017.
- Koustuv Sinha, Shagun Sodhani, Jin Dong, Joelle Pineau, and William L. Hamilton. Clutrr: A diagnostic benchmark for inductive reasoning from text. *Empirical Methods of Natural Language Processing (EMNLP)*, 2019.
- Oscar Skean, Md Rifat Arefin, Dan Zhao, Niket Patel, Jalal Naghiyev, Yann LeCun, and Ravid Shwartz-Ziv. Layer by layer: Uncovering hidden representations in language models. *arXiv preprint arXiv:2502.02013*, 2025.
- Ian Tenney, Dipanjan Das, and Ellie Pavlick. Bert rediscovers the classical nlp pipeline. *arXiv preprint arXiv:1905.05950*, 2019.
- Arnas Uselis and Seong Joon Oh. Intermediate layer classifiers for ood generalization. *arXiv preprint arXiv:2504.05461*, 2025.
- Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. *Advances in neural information processing systems*, 30, 2017.
- Jesse Vig and Yonatan Belinkov. Analyzing the structure of attention in a transformer language model. *arXiv preprint arXiv:1906.04284*, 2019.
- Jesse Vig, Sebastian Gehrmann, Yonatan Belinkov, Sharon Qian, Daniel Nevo, Simas Sakenis, Jason Huang, Yaron Singer, and Stuart Shieber. Causal mediation analysis for interpreting neural nlp: The case of gender bias. *arXiv preprint arXiv:2004.12265*, 2020.
- Cédric Villani et al. *Optimal transport: old and new*, volume 338. Springer, 2008.
- Elena Voita, Rico Sennrich, and Ivan Titov. The bottom-up evolution of representations in the transformer: A study with machine translation and language modeling objectives. *arXiv preprint arXiv:1909.01380*, 2019.
- Zhaofeng Wu, Xinyan Velocity Yu, Dani Yogatama, Jiasen Lu, and Yoon Kim. The semantic hub hypothesis: Language models share semantic representations across languages and modalities. *arXiv preprint arXiv:2411.04986*, 2024.

An Yang, Baosong Yang, Binyuan Hui, Bo Zheng, Bowen Yu, Chang Zhou, Chengpeng Li, Chengyuan Li, Dayiheng Liu, Fei Huang, Guanting Dong, Haoran Wei, Huan Lin, Jialong Tang, Jialin Wang, Jian Yang, Jianhong Tu, Jianwei Zhang, Jianxin Ma, Jin Xu, Jingren Zhou, Jinze Bai, Jinzheng He, Junyang Lin, Kai Dang, Keming Lu, Keqin Chen, Kexin Yang, Mei Li, Mingfeng Xue, Na Ni, Pei Zhang, Peng Wang, Ru Peng, Rui Men, Ruize Gao, Runji Lin, Shijie Wang, Shuai Bai, Sinan Tan, Tianhang Zhu, Tianhao Li, Tianyu Liu, Wenbin Ge, Xiaodong Deng, Xiaohuan Zhou, Xingzhang Ren, Xinyu Zhang, Xipin Wei, Xuancheng Ren, Yang Fan, Yang Yao, Yichang Zhang, Yu Wan, Yunfei Chu, Yuqiong Liu, Zeyu Cui, Zhenru Zhang, and Zhihao Fan. Qwen2 technical report. *arXiv preprint arXiv:2407.10671*, 2024.

Jason Yosinski, Jeff Clune, Yoshua Bengio, and Hod Lipson. How transferable are features in deep neural networks? *Advances in neural information processing systems*, 27, 2014.

Qinan Yu, Jack Merullo, and Ellie Pavlick. Characterizing mechanisms for factual recall in language models. *arXiv preprint arXiv:2310.15910*, 2023.

A Mathematical details for matrix-based information estimation and theoretical foundations

We employ the matrix-based Rényi entropy (Giraldo et al., 2014) to estimate mutual information between representations and labels, leveraging kernel Gram matrices to capture sample similarity.

A.1 Matrix-based entropy estimation

Let $U = \{u_i\}_{i=1}^N \subset \mathbb{R}^d$ denote ℓ_2 -normalized representations obtained from a specific transformer layer, a residual update, or the embedding of the target label. A Gaussian kernel Gram matrix $G_U \in \mathbb{R}^{N \times N}$ is constructed as: $(G_U)_{ij} = \exp\left(-\frac{\|u_i - u_j\|^2}{2\sigma^2}\right)$, with bandwidth $\sigma = 1$. The matrix is then trace-normalized to ensure $\text{tr}(G_U) = 1$.

The matrix-based Rényi entropy of order $\alpha = 1$ is defined as:

$$H(U) = -\text{tr}(G_U \log G_U).$$

This expression can be interpreted in terms of the eigenvalue spectrum $\{\lambda_k\}$ of G_U , since G_U is positive semi-definite and trace-normalized:

$$H(U) = -\sum_{k=1}^N \lambda_k \log \lambda_k.$$

The entropy thus reflects the dispersion of the eigenvalues. A more uniform spectrum (i.e., higher entropy) suggests more diversity in the representation space, while a sharply peaked spectrum (i.e., low entropy) indicates redundancy or compression.

Polynomial Kernel. Given a set of normalized representations $U = \{u_i\}_{i=1}^N$, the polynomial kernel Gram matrix is defined as $(G_U^{\text{poly}})_{ij} = (u_i^\top u_j + c_0)^p$, where p is the polynomial degree and c_0 is a constant bias term. *Laplacian Kernel.* Given the same representation set $U = \{u_i\}_{i=1}^N$, the Laplacian kernel Gram matrix is defined as $(G_U^{\text{lap}})_{ij} = \exp(-\gamma \|u_i - u_j\|_1)$, where $\gamma > 0$ is a kernel coefficient and $\|\cdot\|_1$ denotes the ℓ_1 distance.

A.2 Mutual information estimation

To estimate the mutual information between two random variables U and V , we compute their Gram matrices G_U and G_V , and form the joint similarity matrix via element-wise (Hadamard) product: $G_{UV} = G_U \circ G_V$. After trace-normalization, mutual information is estimated by: $I(U; V) = H(U) + H(V) - H(U, V)$, where $H(U, V) = -\text{tr}(G_{UV} \log G_{UV})$. The eigenvalue spectrum of G_{UV} governs the joint entropy term; its shape reflects how much of the structure in U and V aligns. A more concentrated spectrum in G_{UV} relative to G_U and G_V implies stronger dependence and thus higher mutual information.

B Dataset overview and statistics

Data Statistics We conduct experiments on four datasets with varying levels of complexity and structure: CLUTRR (Sinha et al., 2019), ECQA (Aggarwal et al., 2021), CNN/Daily-Mail (See et al., 2017; Hermann et al., 2015) and a custom-designed Synthetic Arithmetic dataset. Table 3 summarizes key dataset statistics.

B.1 CLUTRR

CLUTRR (Compositional Language Understanding and Text-based Relational Reasoning) (Sinha et al., 2019) is a diagnostic benchmark for evaluating relational inference in

Dataset	#Train	Train Seq. Len	#Val	Val Seq. Len	#Test	Test Seq. Len
CLUTRR	9,074	30	2,020	29	1,146	70
ECQA	7,598	21	1,090	21	2,194	21
CNN/DailyMail	-	-	-	-	11,490	3,967
Synthetic Arithmetic	10,000	9	2,000	9	2,000	9

Table 3: Dataset statistics.

language models. Each example contains a story describing family relations, and the task is to infer the missing relationship between two entities. The distribution shift stems from clause lengths that are absent in the training set but present during evaluation. We use the task split “gen_train23_test2to10”, where the model is trained on clause lengths 2 and 3 and evaluated on lengths 2 through 10.

B.2 ECQA

ECQA (Explanations for CommonsenseQA) (Aggarwal et al., 2021) is a commonsense multiple-choice question-answering dataset, where each question is accompanied by 5 answer options.

B.3 CNN/DailyMail

CNN/DailyMail (See et al., 2017; Hermann et al., 2015) is a summarization dataset containing news articles as written by journalists at CNN and the Daily Mail.

B.4 Synthetic Arithmetic Dataset

We construct a synthetic diagnostic dataset to disentangle task-relevant signal learning from spurious noise memorization in a controlled setting. Each sample consists of a sequence of 10 symbolic elements, where the signal component follows an arithmetic progression modulo K , and the remainder of each element is independently corrupted with random noise. By varying the modulus K , we systematically control task complexity and introduce structured shifts in the data distribution.

Synthetic Arithmetic dataset construction. At each position t , the signal value is computed as:

$$s_t = (s_0 + t \cdot d) \bmod K, \quad \text{with } s_0 \in [0, K-1], \quad d \in [1, K-1].$$

Each element in the sequence is represented as a string of the form:

$$S\{\text{signal}\}.N\{\text{noise}\}, \quad \text{where } \text{noise} \sim \mathcal{U}_{\text{int}}(0, \text{noise_range}-1).$$

Here $\mathcal{U}_{\text{int}}$ denotes the uniform distribution. The model is trained to predict the signal value of the final (10th) element, using the preceding elements as input context.

For example, with $K = 5$, $s_0 = 1$, and $d = 2$, a sample might look like:

S1.N42 S3.N77 S0.N18 S2.N56 S4.N90 S1.N11 S3.N65 S0.N23 S2.N37

Each element encodes both a signal (the number following S) and a noise component (the number following N). The target is 4, corresponding to the signal of the final (10th) item in the sequence.

C Implementation details

This appendix outlines implementation details in our experiments.

C.1 Prompt construction

All tasks are cast into a natural language generation format. The model receives a prompt and generates the next token. Below are construction strategies and examples for each dataset:

CLUTRR Each input example in CLUTRR consists of a short narrative describing a set of family relationships, along with a query involving a pair of entities. We construct prompts by concatenating the narrative and a structured natural language question derived from the query tuple. The model is trained to predict the correct relationship as the next token.

Prompt:

Story: [Alice] is [Bob]’s mother. [Bob] is [Charlie]’s father.
Query: What is the relationship between Alice and Charlie? Answer:

Target: grandmother

ECQA (Explanation-augmented Commonsense QA) Each ECQA instance consists of a multiple-choice question with five candidate answers. We format the prompt by presenting the question followed by all five options (labeled A–E), and conclude with an explicit answer query. The model is trained to predict the correct answer letter as the next token.

Prompt:

Question: What do people usually do at a birthday party?
Options:
A. Sleep
B. Celebrate
C. Cook
D. Exercise
E. Drive
Answer:

Target: B

CNN / DailyMail Each input example consists of an article and its corresponding highlights. We construct the prompt by appending the string `\n\n The following is a one-paragraph highlight summary of the article:\n\n` to the end of the article text.

Prompt:

(CNN)The Palestinian Authority officially became the 123rd member of the International Criminal Court on ...

The following is a one-paragraph highlight summary of the article:

Target: Membership gives the ICC jurisdiction over alleged crimes committed in Palestinian territories since last June . Israel and the United States opposed the move, which could open the door to war crimes investigations against Israelis .

Synthetic Arithmetic Each synthetic sample consists of a sequence of 10 symbolic elements, where each element is formatted as $S\{\text{signal}\}_N\{\text{noise}\}$. The signal values follow an arithmetic progression modulo K , and the noise values are independently sampled from a uniform distribution with a fixed range of 100. During training, the modulus K is set to 13. For evaluation, test sequences are generated using values of K from the range $[5, 26]$ excluding 13 to simulate a distribution shift. The model receives the first 9 tokens as input and is trained to predict the signal component of the 10th token.

Prompt:

S1_N42 S3_N88 S5_N20 S7_N10 S9_N65 S11_N43 S0_N99 S2_N38 S4_N77

Target: 6

This controlled format enables manipulation of distributional properties by varying the modulus K .

C.2 Residual scaling with learnable β_ℓ parameters

We introduce a vector of learnable scalar weights $\beta = \{\beta_1, \dots, \beta_L\}$ applied to residual connections in a frozen transformer:

$$z^{(\ell)} = z^{(\ell-1)} + \beta_\ell \cdot \text{Block}^{(\ell)}(z^{(\ell-1)}).$$

- All transformer weights are frozen; only β_ℓ parameters are trained.
- Each transformer block’s residual output is modulated by β_ℓ via forward hooks.
- β are initialized to 1 and updated using gradient descent without altering the architecture.

For Synthetic Arithmetic, the in-distribution (ID) setting is using $K = 13$, the out-of-distribution (OOD) setting is using $K = [5, 26]$ excluding 13. For CLUTRR, the in-distribution (ID) setting is using the “1.2” and “1.3” split of “gen_train23_test2to10”. Out-of-distribution (OOD) is conducted on held-out CLUTRR configurations that do not overlap with the training distribution: the “1.4” to “1.10” split of “gen_train23_test2to10”. For ECQA, we use training set as the ID setting and CLUTRR as the OOD setting. The ID and OOD sizes are balanced.

C.3 Training and evaluation settings

We finetune the model using HuggingFace’s Trainer API with a batch size of 16, a learning rate of $\alpha = 5 \times 10^{-6}$, and AdamW optimizer with weight decay of 0.01. We perform residual β_ℓ training on top of a model that has already been fine-tuned on the target dataset, keeping all previously learned weights frozen and optimizing only the β_ℓ parameters. We use a batch size of 4, a learning rate of 0.001, Adam optimizer and epochs of 10. Evaluation is conducted by comparing the next predicted token with the next ground-truth target token.

Model	CLUTRR	Synthetic	ECQA
GPT-2 Small	56.57	72.55	—
GPT-2 Medium	62.11	74.05	—
Qwen-2.5-0.5B	58.61	88.00	65.45
LLaMA-3.1-8B	72.62	89.70	80.49

Table 4: Performance (%) of different models across datasets.

D Compute and licensing details

Computing resources Experiments were conducted on an NVIDIA RTX A6000 GPU (48GB).

Model licenses The GPT-2 Small (Radford et al., 2019) and GPT-2 Medium (Radford et al., 2019) models are released under the MIT License and are available via Hugging Face Transformers. The Qwen-2.5 0.5B (Yang et al., 2024) model is provided by Alibaba Group under the Apache 2.0 License. The LLaMA-3.1 8B (Meta AI, 2024) model is made available by Meta under a non-commercial research license and accessed via Hugging Face.

Dataset licenses CLUTRR (Sinha et al., 2019) is released under a CC BY 4.0 license as part of EMNLP 2019. The ECQA (Aggarwal et al., 2021) dataset is also released under a CC BY 4.0 license by its authors as part of EMNLP 2021. CNN/DailyMail is under Apache-2.0 License. The Synthetic Arithmetic dataset is custom-designed by the authors and does not rely on any external or licensed data sources.

All assets were used in compliance with their respective licenses, and no proprietary or restricted resources were employed in our experiments.

E Full quantitative results with confidence estimates

Predictive information To assess estimation variability and sensitivity with the choice of evaluation data, we report the results with statistical significance over 5 random 100 subsamples of test examples.

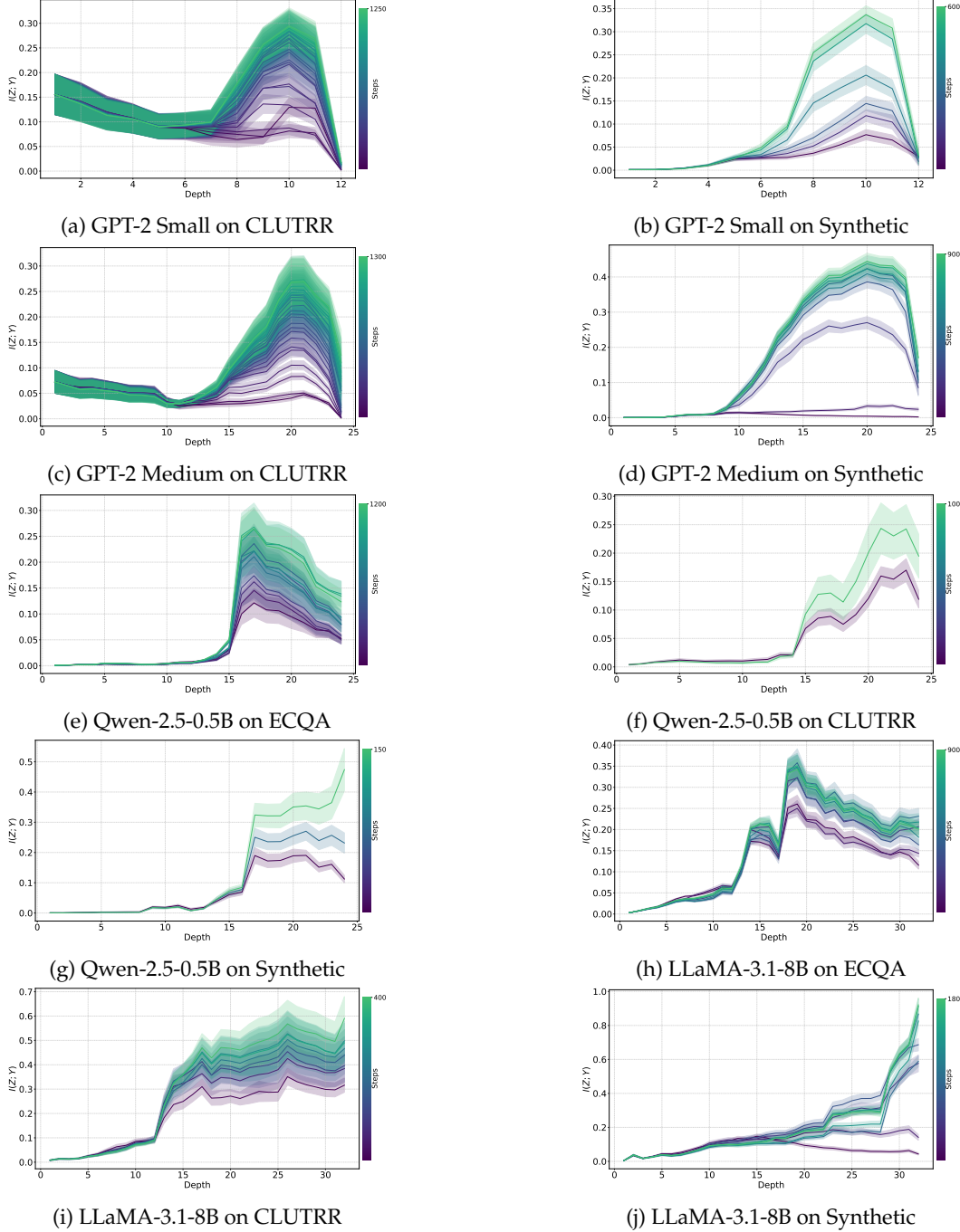


Figure 6: Predictive information $I(Z; Y)$ across models and datasets. Lighter line colors represent later training steps. Each curve shows the mean across 5 sampling seeds (0, 1, 2, 3, 42), and the shaded region denotes a 2-sigma (96%) confidence interval.

Incremental information gain In addition to predictive information $I(Z^{(\ell)}; Y)$, we compute the incremental information gain $I(\Delta Z^{(\ell)}; Y)$ for each transformer block.

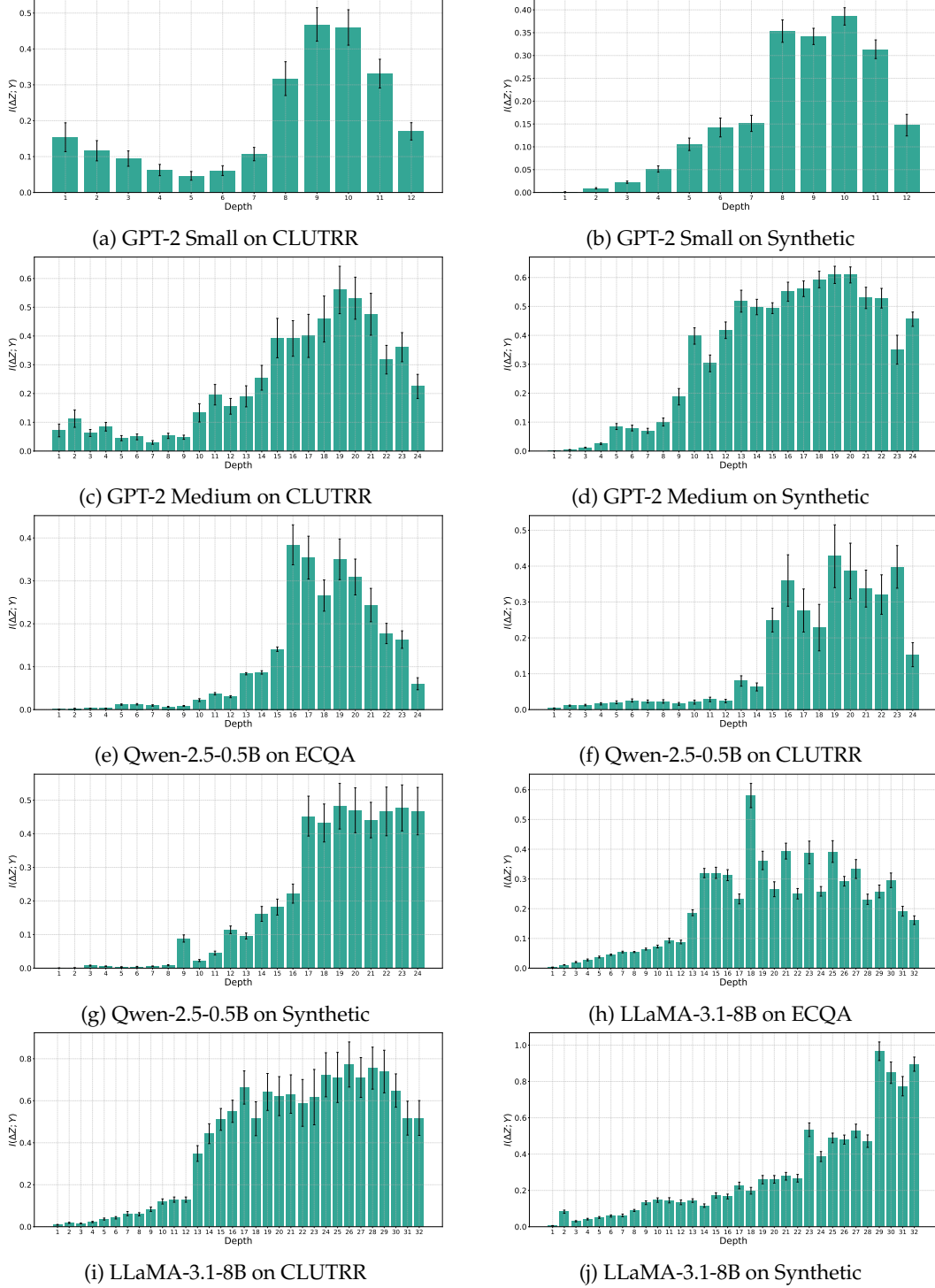


Figure 7: Incremental information gain $I(\Delta Z; Y)$ across different models and datasets with $\sim 96\%$ CI error bars.

Kernel ablation To assess robustness with respect to kernel selection, we additionally test the Laplacian and Polynomial kernels (Figure 8).

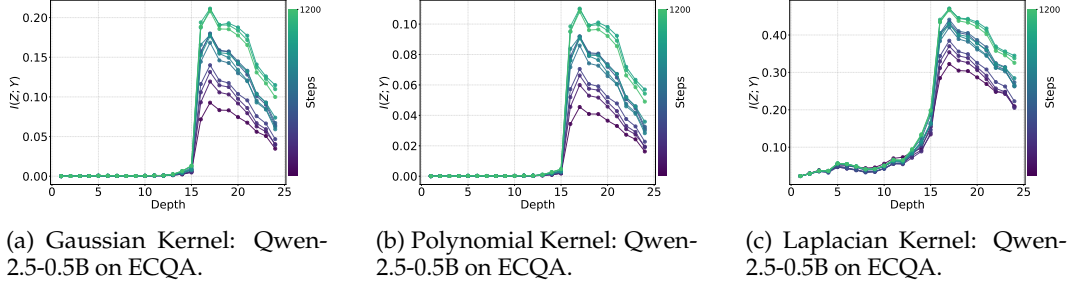


Figure 8: Different kernel show similar pattern.

Residual scaling We present the complete set of residual scaling results, detailing the learned β_ℓ values across all transformer layers.

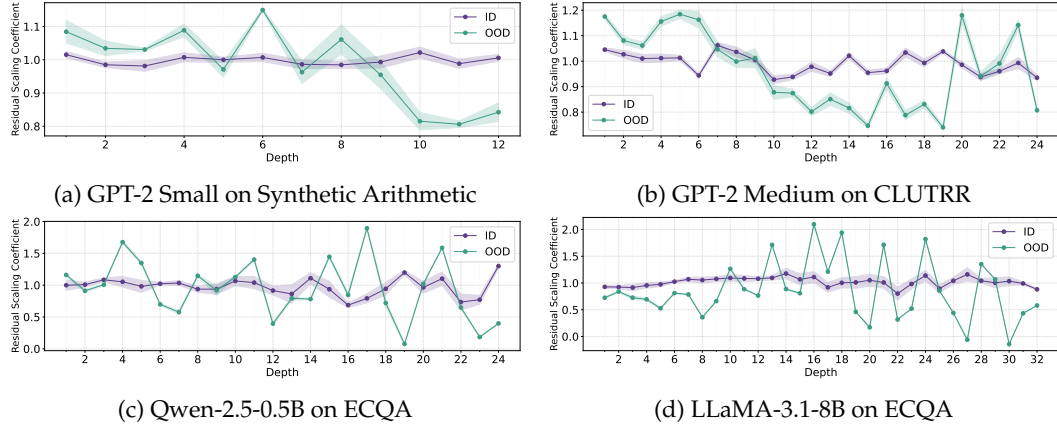


Figure 9: Residual scaling coefficients β_ℓ across all transformer layers. Each curve shows the mean across five random seeds, and the shaded region denotes 1-sigma error bar. ID training emphasizes later layers, while OOD training has lower final layer weight.

Model	Dataset	Before Training		Train on ID		Train on OOD	
		ID	OOD	ID	OOD	ID	OOD
GPT2	Synthetic Arithmetic	100.0 $\pm$ 0.0	39.3 $\pm$ 2.1	100.0 $\pm$ 0.0	39.1 $\pm$ 2.1	71.8 $\pm$ 2.6	65.4 $\pm$ 3.0
GPT2-Medium	CLUTRR	95.8 $\pm$ 0.0	28.7 $\pm$ 3.3	97.9 $\pm$ 0.0	30.4 $\pm$ 4.0	97.9 $\pm$ 0.0	32.7 $\pm$ 3.4
Qwen-2.5-0.5B	ECQA	98.1 $\pm$ 0.3	4.2 $\pm$ 0.0	98.5 $\pm$ 0.2	1.5 $\pm$ 1.8	80.0 $\pm$ 2.2	14.9 $\pm$ 0.3
LLaMA-3.1-8B	ECQA	95.0 $\pm$ 0.4	0.0 $\pm$ 0.0	95.5 $\pm$ 0.4	0.0 $\pm$ 0.0	67.6 $\pm$ 2.7	21.1 $\pm$ 0.5

Table 5: Accuracy (mean $\pm$ std) before and after training on ID or OOD data.

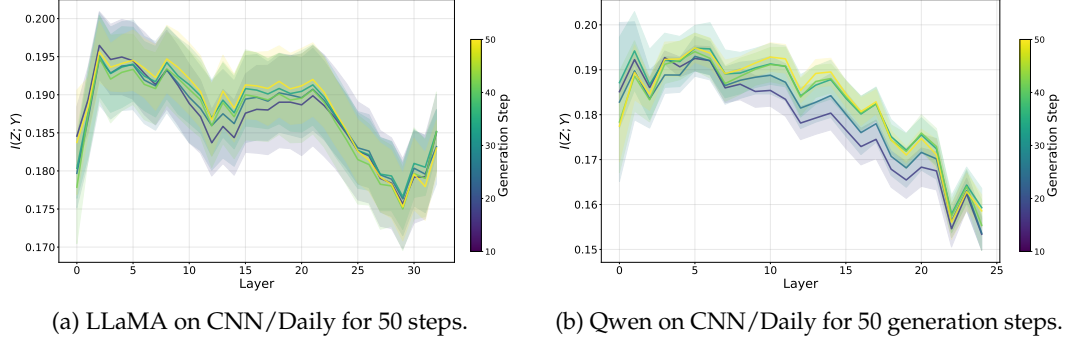


Figure 10: $I(Z_l^{(t)}; Y)$ across layers (x-axis) and generation steps (right color bar), the left y-axis shows mutual information values. Values are plotted every ten steps. Lines denote the mean across three independent runs (seed 42, 123, 1234, temperature 0.7), with shaded regions indicating ± 1 standard deviation.

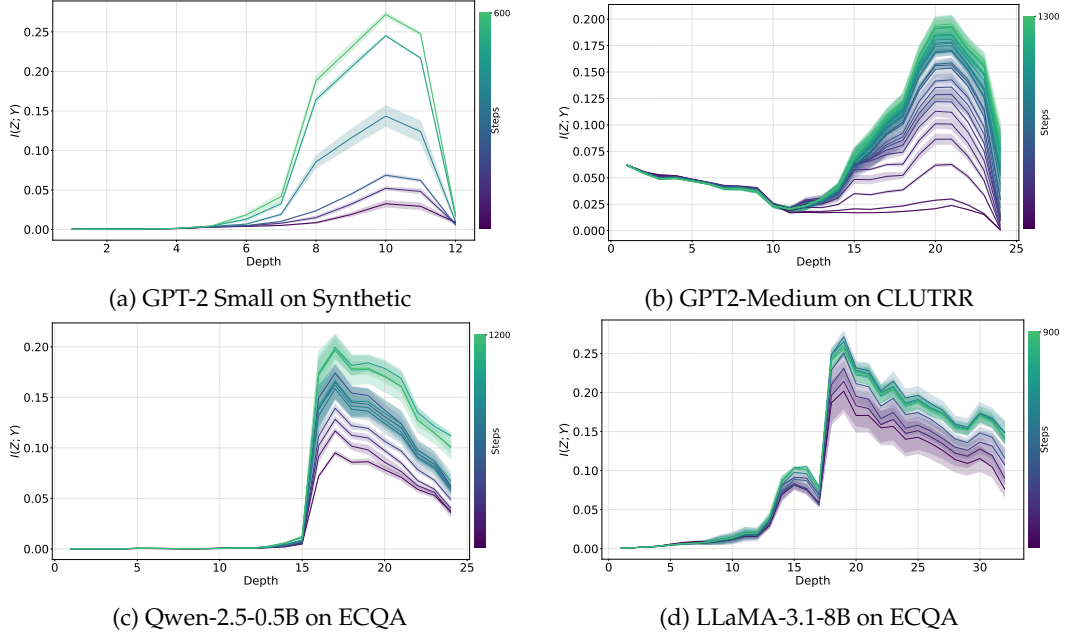


Figure 11: $I(Z_l^{(t)}; Y)$ across layers (x-axis) with standard deviation across 3 different finetuning seed (42, 123, 1234).