ways, formalised as follows:

$$\textbf{Sum:}\quad \alpha([s_i, e_i]) = \sum_{t=s_i}^{e_i} X[t] \tag{2}$$

$$\textbf{Mean:}\quad \alpha([s_i, e_i]) = \frac{1}{e_i - s_i + 1} \sum_{t=s_i}^{e_i} X[t] \tag{3}$$

$$\textbf{Max:}\quad \alpha([s_i, e_i]) = \max_{t\in[s_i,e_i]} X[t] \tag{4}$$

The grouped activations transform as follows:

- For attention layers output, $X \in R^{B\times K\times H_m}$ becomes $X \in R^{B\times G\times H_m}$.
- For FF, $X \in R^{B\times K\times H_f}$ becomes $X \in R^{B\times G\times H_f}$.
- For residual stream:, $X \in R^{B\times K\times H_h}$ becomes $X \in R^{B\times G\times H_h}$.

This process consolidates activations for each syntactic unit, enabling systematic evaluation of compositional robustness across layers. For simplicity, we demonstrate the operation over these components, but this approach can be extended to any transformer's components, provided that the dimensional requirements for information flow, as described in (Elhage et al., 2021), are respected. For example, consider attention layer internal activations of shape $X \in \mathbb{R}^{B\times H_a\times K\times K}$, where H_a is the number of attention heads, and K represents the query and key token dimensions. Applying CAP with the **Sum** protocol involves aggregating activations over the query range $[s_i, e_i]$ and the key range $[s_j, e_j]$. The grouped activations are computed as: $\alpha([s_i, e_i], [s_j, e_j]) = \sum_{t=s_i}^{e_i} \sum_{t'=s_j}^{e_j} X[b, h, t, t']$. After applying CAP, the grouped activations have the shape $X \in \mathbb{R}^{B\times H_a\times G\times G}$, where G is the number of grouped syntactic units. This ensures that query-key interactions are consolidated into cohesive syntactic units, aligning activations with higher-level linguistic structures. We examine CAP's reduction ratio ($K \rightarrow G$) at the word-level and its effects across models, with detailed analysis in Appendix C. We refer the reader to Appendix B.4 for further details on how CAP affects sequence length and interacts with positional encodings.

The CAP effect on models is evaluated by measuring their accuracy post-CAP on a baseline test consisting of examples correctly predicted by the original models. This ensures that the evaluation focuses on instances where CAP directly tests compositional robustness. Specifically, we report three key metrics: the original accuracy (A_o), which represents the model's accuracy on the baseline test before applying CAP and establishes a reference for evaluating the grouping effect; the grouped accuracy (A_c), which measures the model's accuracy post-CAP, averaged across all CAP protocols (sum, mean, max) and reflects how well the model retains its predictions after compositional grouping; and the accuracy drop (ΔA), defined as $\Delta A = A_o - A_c$, which quantifies the performance loss due to CAP, where lower ΔA values indicate more robust compositional behaviour and better preservation of semantic information across layers. These metrics offer a framework for comparing tasks and models, allowing a granular assessment of compositional representations.

4 Empirical analysis

4.1 Experimental setup & datasets

Datasets and metrics. The CAP effect is evaluated using three WordNet-derived datasets—definitions, hypernyms, and synonyms—corresponding to the IDM, HP, and SP tasks (Fellbaum, 1998). Test examples correctly predicted by the original models (A_o) form the baseline for subsequent CAP evaluation. Grouped accuracy (A_c) is calculated post-CAP for this subset, ensuring that CAP's effect is isolated to examples where the original models performed correctly. The drop in accuracy (ΔA) is reported per protocol (sum, mean, max) to assess the impact of different aggregation methods on model performance. See Appendix B.2 for dataset details and Appendix E.3 for comprehensive results.

LLMs and evaluated dimensions. The methodology was tested across various decoder-only transformer models (Vaswani, 2017). Our main focus was on GPT-2 (small: 124M, medium: 355M, large: 774M parameters) (Radford et al., 2019), Gemma1 (2B parameters) (Team et al., 2024), Llama (3B, and 8B parameters) (Dubey et al., 2024), and Qwen (0.5B, 1.5B, and 3B parameters) (Yang et al., 2024). These models use different tokenisation approaches: byte-level BPE (GPT-2, Qwen), expanded BPE with 128K vocabulary (Llama3), and SentencePiece (Gemma). Models were tested before and after task-specific fine-tuning (3 epochs, learning rate 5e-5). This selection spans diverse architectures, sizes, and tokenisation strategies (see

Model	Layer Position	Original			Fine-tuned		
		Max	Mean	Sum	Max	Mean	Sum
GPT2-large	1%	8.06%	9.15%	6.70%	10.61%	10.01%	7.83%
	25%	5.19%	4.94%	5.63%	6.25%	5.77%	6.32%
	75%	5.28%	2.62%	2.39%	3.66%	1.62%	0.88%
	100%	0.84%	0.12%	0.19%	0.22%	0.16%	0.16%
Gemma-2B	1%	97.91%	23.51%	23.75%	57.58%	22.70%	21.99%
	25%	86.32%	16.20%	19.27%	50.45%	14.08%	15.57%
	75%	52.38%	31.03%	24.74%	21.77%	14.99%	12.80%
	100%	6.87%	10.61%	10.61%	2.21%	2.05%	2.05%
Qwen-3B	1%	12.63%	12.27%	11.44%	7.85%	6.71%	6.48%
	25%	18.61%	8.59%	9.11%	10.66%	4.75%	5.82%
	75%	7.23%	4.00%	3.79%	3.65%	2.83%	1.85%
	100%	0.39%	0.4%	0.4%	0.31%	0.17%	0.2%
Llama3-8B	1%	25.49%	24.99%	24.94%	24.44%	23.42%	23.48%
	25%	20.02%	5.87%	5.74%	8.81%	6.03%	5.92%
	75%	7.31%	3.40%	3.54%	5.16%	3.47%	3.29%
	100%	2.80%	1.77%	1.77%	1.55%	1.33%	1.33%

Table 1: IDM accuracy drop Δ in the word-level CAP, highlighting best and worst values in both original and fine-tuned models. The layer numbers were normalised to layer positions as percentages of the total layers, which allows comparing equivalent relative depths across models, such as 25% or 75% of the total layers, rather than using absolute layer numbers. This method ensures fair comparisons between models, even with different architectures.

Appendix B.3 for further details on the models and fine-tuning parameters).

Experimental setup. All experiments were conducted using 2x NVIDIA RTX A6000 and 2x NVIDIA RTX A100 GPUs, with the experimental framework being developed in Python 3.11.5. We used the Transformers (v4.44.2) and PyTorch (v2.4.1) libraries, along with Transformer-lens (v2.6.0), to train and evaluate models and for probing. Benepar (v0.2.0) was used for sentence parsing, and statistical analysis was supported by Scikit-learn (v1.5.2).

4.2 Results and discussion

Compositional inference in LLMs is not a purely incremental process. Contrary to expectations of a smooth and steady layer-wise performance improvement, we observe significant fluctuations when CAP is applied across layers. Performance drops notably in early and middle layers, followed by sharp improvements (Figure 2 (a)-(c), (e), and (f)), suggesting these layers struggle to process CAPed activations, particularly the pooled linguistic features captured in earlier layers. *Rather than progressively building semantic information from individual tokens to complex phrases, the models appear to focus heavily on isolated token features.*

An important distinction arises between **TW-CAP**, which groups tokens according to model-specific tokenisation, and **TP-CAP**, which applies externally parsed syntactic structures. While TP-CAP introduces richer constituent information, it may not align with the model's internal segmentation or syntactic reasoning. This misalignment is not a flaw in CAP, but rather a diagnostic signal: if LLMs encoded human-like syntax, TP-based grouping should be minimally disruptive. The observed drop in performance under TP-CAP suggests that LLMs do not consistently internalise hierarchical syntactic structures. This finding underscores the model's emphasis on local token-level information and supports the conclusions drawn in our information-theoretic analysis.

The results indicate that attention is distributed over input tokens and model layers in a non-systematic and decentralised manner that is highly context-dependent, showing minimal reliance on sequential or positional relationships of constituents. This phenomenon is particularly evident in the sharp decline in SP and HP tasks, where contextual information is limited during phrase-level CAP application. We argue that this behaviour stems from the model's training objective, which maximises information gain in each layer towards predicted tokens at the cost of reducing mutual information between tokens in a single layer. This behaviour means that *aggregation, including syntactic, is performed across multiple layers and thus is not localisable from any single given layer.* An information theoretical analysis elaborates this reasoning in Section 5. Our findings highlight how compositional structures are highly sensitive to token representation dynamics across layers, suggesting that performance fluctuations *can be attributed* *to information loss incurred as a function of token mutual information across layers.*

Larger models are more fragile to compositional perturbations. The IDM task highlights this fragility in larger models, as larger models rely on finer feature extraction. Within families, distinct patterns emerge: original Qwen's smaller variants show better IDM robustness (e.g., at position 25% there was a 7.69% drop on Qwen-1.5B vs 12.11% on Qwen-3B), while Llama3 exhibits capacity-dependent behaviour with the 3B variant being more vulnerable than 8B. Despite having similar reduction ratios to Llama models (see Appendix C), Gemma-2B shows greater sensitivity to perturbations (e.g., at position 1% Max: Gemma-2B drops 97.91% vs. Llama3-8B's 25.49%), likely due to its larger vocabulary enabling finer-grained tokenisation. While fine-grained token knowledge benefits standard tasks, it appears to increase susceptibility to compositional perturbations. The superior performance of Llama3-8B over its 3B variant can be attributed to its enhanced capacity for maintaining feature relationships across layers while preserving key compositional information. While larger models excel in standard tasks (see Appendix E.1), *they exhibit a greater reliance on the identification of intrinsic features in the early layers.* We find that phrasal-level CAP substantially impacts Gemma-2B and Llama models, suggesting a heavy dependence on layer-wise information gain, where they separate features in an uncorrelated and highly distinct manner. While this aids in identifying complex feature patterns, it also makes them more vulnerable to contextual noise—*a weakness that threatens their robustness and integrity*. Notably, Qwen models outperform Llama and Gemma despite similar parameter counts, likely due to byte-level BPE tokenisation and multilingual training, which enhance compositional stability, whereas Llama's expanded BPE and Gemma's SentencePiece prioritise efficiency over phrase retention, increasing vulnerability to CAP interventions.

Activation abstraction vs the information loss. Table 1 reveals significant variations in aggregation function performance across sample models for the IDM task (see Appendix E.3 for the rest of the models and tasks results). The Max aggregation shows the most dramatic impact. *This finding supports our argument that these models tend to distribute information in a fragmented manner, lacking the integration of compositional (lexical and semantic) information across tokens and contiguous layers.* The Mean aggregation provides more balanced results, though performance drops still indicate *absence of consistent compositional mechanisms.* This issue becomes more pronounced in token-phrases experiments (Figure 2). The *Sum aggregation consistently outperformed other methods*, with Mean aggregation following closely behind, particularly in original models. The Sum aggregation reflects the cumulative effect of aggregating tokens into larger segments, reinforcing our earlier conclusion. Instead of progressively building semantic information across layers, *the models exhibit cumulative information loss,* particularly when interventions occur in early layers.

Fine-tuning enhances recovery capabilities across models. Figure 2 (d-f) demonstrates improved performance maintenance post-fine-tuning across all model families, with strongest gains in 75%-100% layer positions. SP tasks showed maximum benefit, attributed to high task specificity and minimal activation reduction under CAP. Max aggregation displayed the greatest improvement post-fine-tuning, likely due to enhanced retention of key information. For instance, Gemma-2B's accuracy drop decreased from 97.91% to 57.65% in the 1% layer, while Qwen-3B improved from 7.23% to 3.65% in the 75% layer. Mean aggregation benefits were also substantial in smaller models, with Gemma-2B's 75% layer drop reducing from 31.03% to 15.00%. The Qwen family showed consistent improvements across all aggregation types, though smaller models like GPT2-large demonstrated minimal gains, suggesting potential overfitting. Notably, larger models like Llama3-8B showed minimal gains from fine-tuning in IDM tasks, indicating that standard fine-tuning objectives may not directly enhance compositional robustness. Although fine-tuning strengthens models' resilience under CAP, it does not fully resolve the challenge of forming stable compositional semantic representations, highlighting an architectural limitation in current transformer models.

5 Information Gain & Token Mutual Information

The empirical findings can be explained by looking at the autoregressive next-token objective of a transformer model from an information theoretical standpoint: examining the relationship between each generated token Y to the input token representations $R_l(X)$ of each layer l, in terms