CARMA: Enhanced Compositionality in LLMs via Advanced Regularisation and Mutual Information Alignment

Anonymous ACL submission

Abstract

Large language models (LLMs) struggle with compositional generalisation, limiting their ability to systematically combine learned components to interpret novel inputs. While architectural modifications, fine-tuning, and data augmentation improve compositionality, they often have limited adaptability, face scalability constraints, or yield diminishing returns on real data. To address this, we propose CARMA, an intervention that enhances the stability and robustness of compositional reasoning in LLMs while preserving fine-tuned performance. CARMA employs mutual information regularisation and layer-wise stability constraints to mitigate feature fragmentation, ensuring structured representations persist across and within layers. We evaluate CARMA on inverse dictionary modelling and sentiment classification, measuring its impact on semantic consistency, performance stability, and robustness to lexical perturbations. Results show that CARMA reduces the variability introduced by fine-tuning, stabilises token representations, and improves compositional reasoning. While its effectiveness varies across architectures, CARMA's key strength lies in reinforcing learned structures rather than introducing new capabilities, making it a scalable auxiliary method. These findings suggest that integrating CARMA with fine-tuning can improve compositional generalisation while maintaining task-specific performance in LLMs.

1 Introduction

Compositional generalisation (CG) refers to the ability to systematically combine known expressions to generate novel ones following learned rules (Partee, 1984). This capability is essential for advancing language models (LMs) towards robust linguistic understanding beyond mere pattern matching (Ram et al., 2024).

Despite their strong performance across various NLP tasks, large language models (LLMs) exhibit persistent weaknesses in compositional generalisation (Hupkes et al., 2020; Kim and Linzen, 2020a; Aljaafari et al., 2024). These limitations stem from multiple factors, including training objectives and model architectures. Standard autoregressive training methods, such as next-token prediction, prioritise statistical correlations in token sequences over structured semantic understanding (Yin et al., 2023a; Dziri et al., 2024). As a result, token representations often lack structured compositionality, leading to fragmented information processing within layers (horizontal misalignment) and across layers (vertical inconsistency).

Additionally, while self-attention mechanisms in Transformer models effectively capture local dependencies, they frequently fail to maintain coherent compositional representations across multiple layers (Murty et al., 2023). This misalignment impairs the model's ability to generalise compositionally, resulting in sensitivity to input order (Ismayilzada et al., 2024) and difficulties in handling complex syntactic and morphological structures (Aljaafari et al., 2024).

Several approaches have been proposed to address these limitations, including architectural modifications, enhanced encoding strategies, and targeted regularisation techniques (Ontanon et al., 2022; Murty et al., 2023; Csordás et al., 2021). However, these methods often struggle to balance compositional improvements with maintaining performance across diverse downstream tasks. Moreover, their effectiveness is typically confined to specific compositional structures or synthetic benchmarks. Developing a robust and adaptable solution that enables LLMs to achieve consistent CG across diverse tasks remains a major challenge.

This work introduces **CARMA**: enhanced **C**ompositionality in LLMs via **A**dvanced **R**egularisation and **M**utual Information **A**lignment, illustrated in Figure 1. CARMA enhances CG by addressing training challenges that hinder struc-

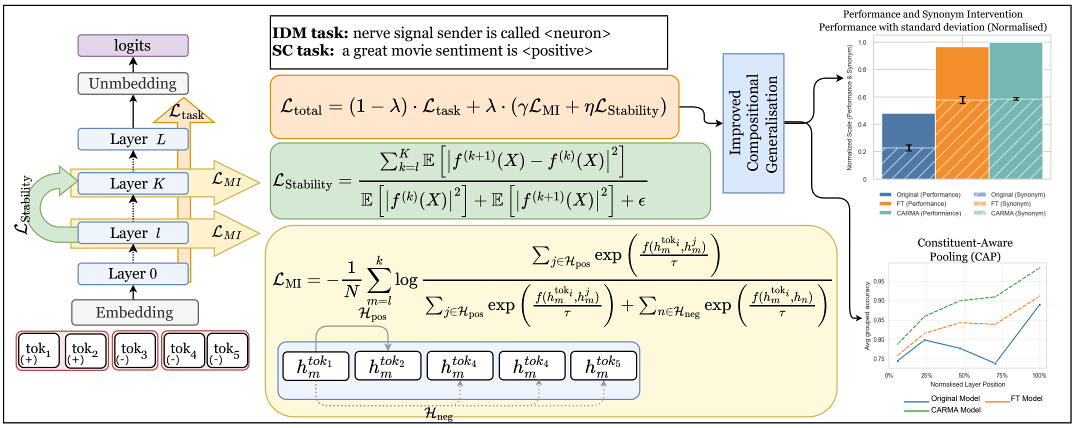


Figure 1: This diagram depicts the computation of the loss and illustrates the integration of the **Mutual Information (MI) loss** ($\mathcal{L}_{\text{MI}}$) and the **Stability Loss** ($\mathcal{L}_{\text{stability}}$) into the final optimisation process. Tokens Tok_1 and Tok_2 form the *positive set* (H_{pos}), while Tok_3, Tok_4, Tok_5 form the *negative set* (H_{neg}). The $\mathcal{L}_{\text{MI}}$ loss is computed *vertically* across layers (l to k), maximising the similarity of tokens in H_{pos} while contrasting them with tokens in H_{neg}. The $\mathcal{L}_{\text{stability}}$ loss is computed *horizontally* between consecutive layers, ensuring consistency in hidden state representations. Both auxiliary losses are combined with the task loss ($\mathcal{L}_{\text{task}}$) to form the total loss ($\mathcal{L}_{\text{total}}$). This integration improves token representations and enhances the model's overall optimisation.

tured compositionality in LLMs. By balancing layer-specific updates and reinforcing token-level dependencies, CARMA provides a scalable and adaptable solution that improves CG without sacrificing downstream task performance. To evaluate CARMA's effectiveness, we investigate the following research questions:

- **RQ1:** How does regulating mutual information across layers influence compositionality in LLMs? How does it affect sensitivity to input and internal perturbations?

- **RQ2:** To what extent does layer-specific regularisation improve compositional generalisation across semantic and sentiment analysis tasks, assessing CARMA's adaptability across domains?

The key contributions of this work are as follows:

- A novel regularisation method that enhances compositional generalisation without requiring architectural modifications. CARMA leverages mutual information alignment to preserve token dependencies across layers and employs layer-wise stability constraints to reduce representational inconsistencies.

- A systematic evaluation of CARMA across compositionally demanding tasks, demonstrating its ability to reinforce systematicity and substitutivity, particularly in models where fine-tuning alone is insufficient.

- A theoretical and empirical analysis of how token dependencies degrade across layers in standard LLMs, revealing that CG limitations are not solely dependent on model size but rather on representational instability. CARMA mitigates this by ensuring consistent information flow, showing that non-intrusive regularisation strategies can significantly improve CG.

The remainder of this paper is structured as follows: Section 2 reviews compositionality in LLMs and associated challenges. Section 3 introduces the CARMA method. Section 4 describes the experimental setup. Section 5 presents empirical findings. Section 6 discusses related work. Section 7 offers insights and future research directions. Supporting datasets and software are available at a public repository.[1]

2 Compositionality in LLMs

Compositional generalisation (CG) in linguistics encompasses five key principles: systematicity, productivity, substitutivity, localism, and overgeneralisation (Dankers et al., 2022a). These principles have been explored in LLMs across compositional instruction (Yang et al., 2024b), semantic

[1] Anonymised for review.

parsing (Li et al., 2023), translation (Li et al., 2021), and multi-step inference (Zhang et al., 2024). Studies show standard Transformer-based LLMs exhibit limited CG, struggling with basic compositional tasks such as assembling tokens into words or constructing morphemes (Aljaafari et al., 2024; Ismayilzada et al., 2024). These limitations are linked to architectural constraints, training objectives, and tokenisation practices that fragment information and increase sensitivity to input order and contextual noise (Murty et al., 2023).

Training Objectives and Information Fragmentation. Standard training objectives for LLMs typically optimise for next-token prediction, which prioritises surface-level correlations over deeper semantic integration (Dziri et al., 2024). While this approach is effective for data already seen, it often impedes CG by reducing mutual information between dependent tokens, thereby limiting the model's ability to form coherent compositional representations (Aljaafari et al., 2024).

Architectural Mechanisms and Compositional Consistency. Beyond training objectives, architectural mechanisms such as dropout and self-attention disperse information across the model, increasing sensitivity to input order and context. This undermines **compositional consistency** (Sajjadi et al., 2016; Cai et al., 2021), the ability to maintain consistent outputs when processing semantically equivalent inputs through transformations like word substitution or paraphrasing. These challenges impact both high-complexity reasoning tasks and simpler operations that demand consistent morphological and syntactic processing (Ismayilzada et al., 2024).

Existing Approaches to Enhance CG in LLMs. Research has explored architectural adjustments, regularisation techniques, and task-specific strategies to address CG limitations. Ontanon et al. (2022) demonstrated that combining relative positional encoding with embeddings enhances CG in algorithmic tasks, while weight sharing and copy decoders help retain input structures. Architectural modifications like Pushdown Layers (Murty et al., 2023) and GroCoT (Sikarwar et al., 2022) incorporate mechanisms for tracking syntactic depth and spatial relations, enabling recursive processing of compositional structures. RegularGPT (Chi et al., 2023) introduces adaptive depth and memory mechanisms to facilitate CG. Studies by Csordás et al. (2021) and Petty et al. (2024) reveal that architectural choices and training setups significantly impact CG enhancement. In neural machine translation, Dankers et al. (2022b) found a positive correlation between data size and compositional performance.

Frameworks like CompMCTG and Meta-MCTG (Zhong et al., 2024) suggest joint training and meta-learning approaches improve fluency, though performance drops persist in out-of-distribution tasks. Synthetic tasks show recursive, step-by-step prompt formats support combinatorial generalisation, despite training biases and sequence order constraints (Ramesh et al., 2024).

3 Enhanced Compositionality via Advanced Regularisation and Mutual Information Alignment (CARMA)

This section formalises compositionality, introduces the core principles of CARMA, and details its components. Figure 1 illustrates the method, highlighting its process and key components.

3.1 Compositionality Formalisation

Mathematical Foundations of Compositionality. CG (Section 2) can be formally defined through a compositional system where $\mathcal{E}$ denotes a set of expressions (e.g., token sequences recognised by the model), and $\mathcal{M}$ represents a corresponding set of meanings. This relationship is formalised as a function:

$$f : \mathcal{E} \to \mathcal{M} \tag{1}$$

For any complex expression $e \in \mathcal{E}$, composed of constituent elements $e_1, \ldots, e_n$ according to a syntactic rule r, the function f satisfies:

$$f(r(e_1, \ldots, e_n)) = g_r(f(e_1), \ldots, f(e_n)), \tag{2}$$

where g_r is the semantic operation that corresponds to the syntactic rule r.

Compositional Generalisation in LLMs. Effective CG in LLMs requires generating structured compositions that preserve semantic consistency. Given a novel expression e_{novel} similar to a known expression e_{known} within a threshold β, their semantic functions must remain within an interpretable bound or deviation α:

$$d(e_{\text{novel}}, e_{\text{known}}) \leq \beta \Rightarrow d(f(e_{\text{novel}}), f(e_{\text{known}})) \leq \alpha. \tag{3}$$

This formulation captures systematicity (structured combinations), substitutivity (preservation under transformations), and resistance to overgeneralisation (bounded semantic deviation) while maintaining interpretability.

3.2 CARMA Formalisation

CARMA operates over a range of target layers, from l to $\mathcal{K}$ ($0 < l \leq \mathcal{K} \leq L$, where L is the total number of layers), and consists of two core components: **Mutual Information** and **Layer-Wise Stability** Regularisation.

Mutual Information (MI) Regularisation Across Layers. CARMA preserves essential dependencies and maintains structural coherence by maximising MI between hidden states of related tokens. The MI between hidden states h_i^k and h_j^k at layer k, representing two related tokens i and j, is defined as:

$$I(h_i^k; h_j^k) = \mathbb{E}_{P(h_i^k, h_j^k)} \left[\log \frac{P(h_i^k, h_j^k)}{P(h_i^k)P(h_j^k)} \right] \quad (4)$$

Since exact computation is intractable, MI is approximated using the InfoNCE loss (Oord et al., 2018), encouraging token-level dependencies across the same layers:

$$\begin{aligned} \mathcal{L}_{\text{MI}} = -\frac{1}{N} \sum_{k=l}^{\mathcal{K}} \sum_{i=1}^{Q} \Bigg(\log \sum_{\substack{h_j \in \mathcal{H}^k \\ j \neq i}} \exp\left(\frac{f(h_i^k, h_j^k)}{\tau}\right) \\ - \log \Bigg(\sum_{\substack{h_j \in \mathcal{H}^k \\ j \neq i}} \exp\left(\frac{f(h_i^k, h_j^k)}{\tau}\right) \\ + \sum_{h_m \in \mathcal{N}^k} \exp\left(\frac{f(h_i^k, h_m)}{\tau}\right) \Bigg) \Bigg), \end{aligned} \quad (5)$$

where $f(h_i^k, h_j^k)$ is a similarity function quantifying the relationship between hidden states at layer k, $\mathcal{H}^k$ denotes the set of positive examples related to h_i^k, $\mathcal{N}^k$ is the set of negative examples unrelated to h_i^k at layer k, τ is the temperature parameter, and N is the total number of target layers from l to $\mathcal{K}$, with Q representing the number of tokens or samples used per layer. Further details on MI approximation are provided in Appendix D.

Layer-Wise Stability Regularisation. This component enforces smooth transitions across layers, reducing abrupt changes that could disrupt compositional structures. For a layer k, the Layer-Wise Stability Loss is defined as:

$$\mathcal{L}_{\text{Stability}} = \sum_{k=l}^{\mathcal{K}} \mathbb{E} \left[\frac{\left| f^{(k+1)}(X) - f^{(k)}(X) \right|^2}{\mathbb{E}\left[|f^{(k)}(X)|^2 \right] + \mathbb{E}\left[|f^{(k+1)}(X)|^2 \right] + \epsilon} \right], \quad (6)$$

where $f^{(k)}(X)$ denotes the output of layer k (i.e., after the attention and MLP submodules), and ϵ is a small positive constant to ensure numerical stability (e.g., $\epsilon = 10^{-8}$). Minimising this loss preserves compositional integrity across the specified layers by encouraging smooth and consistent transitions between them, thereby enabling more stable information flow and aggregation within this range.

CARMA Loss. CARMA integrates $\mathcal{L}_{\text{MI}}$ and $\mathcal{L}_{\text{Stability}}$ into its total loss as:

$$\mathcal{L}_{\text{CARMA}} = \gamma \mathcal{L}_{\text{MI}} + \eta \mathcal{L}_{\text{Stability}}, \quad (7)$$

where γ and η are hyperparameters in $[0, 1]$ that control the relative contribution of each component. The final optimisation objective balances task-specific performance with CARMA's regularisation as:

$$\mathcal{L}_{\text{total}} = (1 - \lambda) \cdot \mathcal{L}_{\text{task}} + \lambda \cdot \mathcal{L}_{\text{CARMA}}, \quad (8)$$

where $\mathcal{L}_{\text{task}}$ represents the task-specific loss, $\mathcal{L}_{\text{CARMA}}$ is the regularisation loss, and $\lambda \in [0, 1]$ controls the trade-off between task accuracy and compositional robustness.

Layer Selection for Regularisation. We apply CARMA to layers around one-third of the model depth, based on evidence that early-to-mid layers better capture compositional and syntactic structure, while deeper layers tend to specialise in task-specific representations (He et al., 2024; Langedijk et al., 2024). In our preliminary experiments, we observed that performance gains diminish when regularisation is applied to deeper layers. As a default, we recommend layers 3–4 in 12-layer models and 6–10 in 24-layer models.

Evaluation Metrics. We use exact match accuracy as the primary metric for both IDM and SC. This choice is motivated by the fact that both tasks have closed and categorical output spaces: IDM outputs are limited to a predefined set of single-word lexical entries from WordNet, while SC uses a fixed sentiment label set. In such settings, exact match is a standard and appropriate evaluation criterion.

4 Experimental Setup

4.1 Downstream Tasks & datasets

Two tasks that assess different aspects of compositional generalisation are used in the paper: **Inverse Dictionary Modelling (IDM)** for word-level composition and **Sentiment Classification (SC)** for phrase-level structure. These tasks measure systematicity, substitutivity, over-generalisation, and robustness to perturbations.

IDM evaluates a model's ability to generate terms from definitions, focusing on substitutivity in semantic composition. Using WordNet (Miller, 1994) with an 80-10-10 train-validation-test split, models are prompted with a definition to generate the corresponding term (e.g., The shore of a sea is called" → coast"). By mapping definitions to terms, this task provides a robust assessment of a model's ability to perform compositional substitution.

SC assesses the model's ability to infer sentiment from phrases and sentences, particularly focusing on sentiment shifts and over-generalisation. Using the Stanford Sentiment Treebank (SST) (Socher et al., 2013) with its original splits, models predict sentiment labels from textual inputs (e.g., A brilliant performance sentiment is" → positive"). This task examines how sentiment composition is preserved across different levels of linguistic structure. For both tasks, performance is assessed using Exact Match Accuracy, providing a robust assessment of compositional substitution ability. Task formalisation, dataset details, and task selection rationale are in Appendices A, B.1, and B.2, respectively.

4.2 Models and Experimental setup

We evaluate three setups: original models, task-specific fine-tuning, and fine-tuning with CARMA regularisation. We test GPT-2 (S/L) (Radford et al., 2019), Gemma1–2B (Team et al., 2024), LLaMA3.2 (1B/3B) (Dubey et al., 2024), and Qwen2.5 (0.5B/3B) (Yang et al., 2024a). We focus on decoder-only architectures, as they represent the dominant paradigm in many open-weight and production-ready LLMs. CARMA is generally applied at approximately one-third of the model's depth, though specific layer positions vary. Details on software, FT methodologies, model specifications, and CARMA hyperparameter selection are provided in Appendices B.3 and B.5.

4.3 Interventions for Compositional Robustness and Performance Stability

Two interventions are used to evaluate the robustness of compositional structures and the stability of learned representations: Constituent-aware pooling (CAP) and synonym replacement. These interventions assess hierarchical dependencies and semantic consistency under controlled perturbations.

CAP (Aljaafari et al., 2024) groups token-level representations into higher-level semantic units (e.g., words, constituents) to assess hierarchical dependencies and how compositional structures are maintained across layers. In this paper, the token-to-word CAP is utilised. Model robustness is measured by monitoring performance metrics before and after applying CAP. Full methodology and formalisation are provided in Appendix C.1.

Synonym Replacement evaluates semantic consistency by substituting 25% and 40% of prompt words with synonyms within an interpretable bound (α). Experiments were repeated at least five times with different seeds for robustness and performance stability assessment; further details are in Appendix C.2.

5 Results and discussion

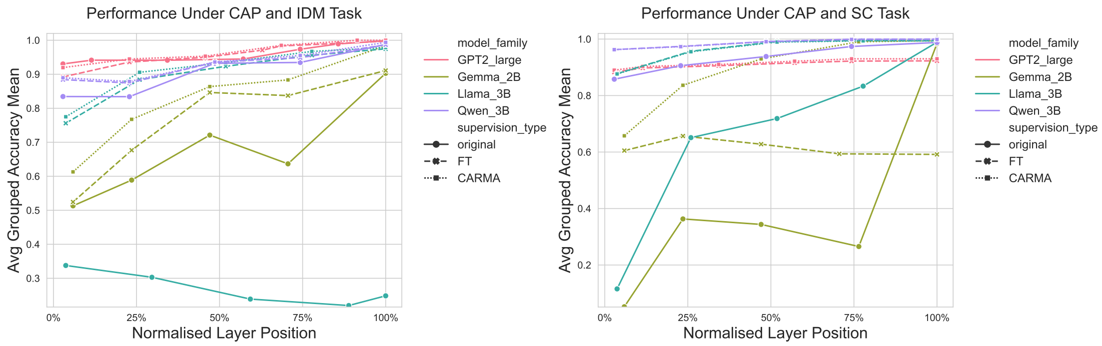
The method is evaluated across three aspects its impact on: (1) model robustness against compositional-based perturbations, (2) model performance stability, and (3) model overall performance. See Appendix B.4 for a detailed breakdown of the evaluation metrics used for each aspect.

5.1 Constituent-Aware Pooling (CAP) Intervention

Fig. 2(a) and 2(b) show the impact of CAP on both tasks, comparing original, fine-tuned (FT), and CARMA models.[2] Each plot shows performance across normalised layer positions, where Accuracy is averaged over three CAP protocols (Mean, Max, Sum); protocol-specific results and extended comparisons are included in Appendix E. The analysis examines how well models preserve compositionality under hierarchical pooling.

CARMA's effectiveness is influenced by model size, tokenisation strategy, and task complexity. In IDM tasks, CARMA models have considerable gains when applying CAP at the earliest layers (1% of model depth), particularly in models with fine-grained tokenisation: Llama-1B (+3.61%) and Gemma-2B (+16.89%). GPT2-L, despite its reliance on subword tokenisation, benefits from CARMA over FT (+3.67%). However, Llama-3B and Qwen-3B minimal improvements (+1.0%) suggest a capacity ceiling where increased model size does not yield proportional gains due to training data limitations. The combination of smaller scale and multilingual training particularly affects Qwen-

[2] Throughout this paper, models incorporating CARMA with FT are referred to as CARMA models.



(a) IDM Task (b) SC Task

Figure 2: Layer-wise performance comparison under CAP intervention, with performance averaged over three protocols (Mean CAP, Max CAP, Sum CAP) for Original, Fine-Tuned (FT), and CARMA (FT + CARMA) models. Layer numbers are normalised to their relative positions within each model to enable cross-architecture comparison. The IDM task (left) highlights CARMA's improvements in systematicity and stability, particularly in the early and middle layers. The SC task (right) demonstrates CARMA's ability to enhance robustness, though convergence with FT occurs in deeper layers.

0.5B, where limited model capacity coupled with broad language coverage appears to constrain English-specific compositional learning, resulting in reduced CARMA benefits. In SC tasks, tokenisation effects vary with task complexity. When intervening at 25% layer position, Gemma-2B and Llama-1B show the strongest gains (+27.38%, +10.59%), while Llama-3B exhibits a marginal difference between CARMA and FT ($\sim 1\%$) but still outperforms the Original model (+37.68%). These results suggest that fine-tuning alone is sufficient for simpler tasks, whereas structured interventions like CARMA are particularly beneficial for more complex, compositional reasoning tasks.

In a layer-wise analysis, the impact of CARMA varies significantly across network depths, revealing crucial insights about compositional learning in transformers. Early layers (0-25%) benefit the most from regularisation, as they establish foundational compositional representations by exhibiting a weak notion of compositionality. Middle layers (25-75%) reinforce these patterns, maintaining structured feature dependencies with moderate improvements. Deeper layers (75-100%) show minimal benefits as the model transitions from compositional learning to task-specialised representations. This pattern aligns with previous findings on layer-wise compositional evolution in Transformers, where earlier layers capture hierarchical structure, while deeper layers exhibit increased task specificity (Feucht et al., 2024). CARMA can thus be strategically applied to control these early representations, maintaining beneficial compositional structure while allowing natural task-specific adaptations in deeper layers.

These findings demonstrate CARMA's effectiveness, particularly for models with granular tokenisation under data constraints, mediated by model capacity and task demands. The method's dual role - enhancing early compositional learning while preserving deeper layer adaptations - enables targeted improvement in model robustness without disrupting task-specific processing.

Model	Ver.	Task	Int.	CS	CV
GPT2-L	CARMA	IDM	25%	56.31	**0.0164**
	FT	IDM	25%	**56.95**	0.0311
	Org	IDM	25%	51.10	0.1175
	CARMA	SC	25%	**0.8858**	**0.0065**
	FT	SC	25%	0.8804	0.0082
Gemma-2B	CARMA	IDM	25%	56.70	**0.023**
	FT	IDM	25%	**57.42**	0.030
	Org	IDM	25%	49.47	0.031
	CARMA	SC	25%	78.90	**0.008**
	FT	SC	25%	**80.23**	0.009
	Org	SC	25%	68.14	0.042
Llama-3B	CARMA	IDM	25%	**62.86**	**0.015**
	FT	IDM	25%	62.22	0.029
	Org	IDM	25%	52.47	0.035
	CARMA	SC	25%	84.83	**0.0056**
	FT	SC	25%	**85.85**	0.0065
	Org	SC	25%	35.21	0.0136

Table 1: Model performance (25% synonym intervention). **Ver.**: Version; **Int.**: Intervention rate; **CS**: ConsistSyn (%); **CV**: Coefficient of Variation. **Best values in bold.**

5.2 Synonyms Replacement Intervention

Synonym Replacement evaluates semantic consistency and robustness under lexical variations across multiple runs ($N \geq 5$) with different seeds. *ConsistSyn* measures output preservation after substitution, while the coefficient of variation (CV) quantifies performance stability, with lower values indicating higher stability. Performance is assessed at 25% and 40% word replacement rates to measure sensitivity to perturbations. Sample results are in Table 1; full details appear in Appendix E.

Across models, CARMA achieves a distinctive performance profile, matching or exceeding FT *ConsistSyn* while consistently demonstrating superior stability through lower CV values. At 25% intervention, Gemma-2B CARMA achieves 56.70% *ConsistSyn* with a CV of 0.0225, compared to FT's 57.42% with higher variance (CV: 0.0307). Llama-3B CARMA outperforms FT in both *ConsistSyn* (62.86% vs. 62.22%) and stability (CV: 0.0148 vs. 0.0292) for IDM. Qwen-3B follows a similar trend but with smaller relative gains, improving stability (CV: 0.0225 vs. 0.0279) while maintaining a marginal *ConsistSyn* advantage over FT (62.00% vs. 61.79%). However, as intervention complexity increases to 40%, the performance gap widens; for example, Gemma-2B FT maintains higher *ConsistSyn* (44.98%) than CARMA (42.36%), though CARMA remains more stable (CV: 0.0174 vs. 0.0249). This behaviour implies that the advantage of CARMA lies in its lower variance and reinforcement of compositional consistency. Thus, it maintains compositional understanding without sacrificing performance, whereas FT produces a performance-driven approach.

While the absolute differences in *ConsistSyn* between CARMA and FT are sometimes modest, particularly at lower replacement rates (e.g., 25%), the stability benefits of CARMA become more evident under increased perturbation (e.g., 40%), where FT models often show degraded consistency. *In these higher-variance regimes, CARMA consistently reduces output variability across model families, reinforcing its utility as a robustness-oriented intervention, even when raw accuracy remains comparable.*

The tokenisation method significantly affects CARMA's impact. Models with more structured tokenisation show stronger stability improvements, but gains vary based on vocabulary design and language coverage. Llama and GPT2-L generally benefit more than Qwen, even with similar sizes, likely due to their smaller multilingual coverage, which results in a more compact and consistent token distribution. Qwen, with a larger vocabulary (151K tokens) supporting broader multilingual processing, introduces redundancy that dampens CARMA's relative stability advantage. Gemma-2B, optimised for a single dominant language with a large vocabulary size, shows the highest overall gains, reinforcing that a structured tokenisation approach focused on a limited linguistic scope enhances CARMA's effectiveness.

Task complexity further differentiates CARMA's effect. CARMA's advantages align with its methodological design, particularly in tasks requiring explicit structural reinforcement. In IDM, where systematicity and substitutivity are critical, CARMA ensures structured mappings hold under perturbation, particularly in Gemma-2B (+14.6% over the original) and Llama-1B (+2692.5% over the original in SC). However, in SC, where compositionality is more distributed, larger models show lower differences between CARMA and FT, reinforcing that larger models encode sentiment shifts effectively without additional intervention.

These results strengthen the hypothesis that CARMA enhances model robustness across perturbations, particularly in structured learning tasks and models where fine-tuning alone does not fully capture compositional dependencies. While FT maintains an advantage in absolute accuracy, CARMA ensures greater consistency, making it critical for improving compositional alignment and mitigating instability in high-variance settings.

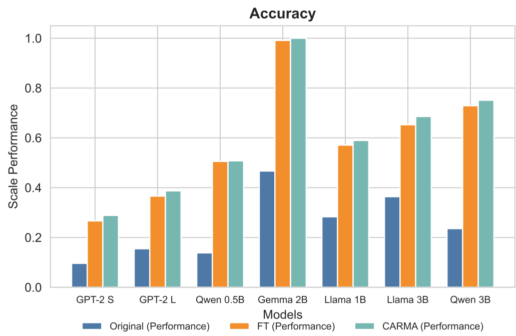


Figure 3: Task performance in IDM across GPT2 (S, L), Gemma-2B, Llama (1B, 3B), and Qwen (0.5B, 3B).

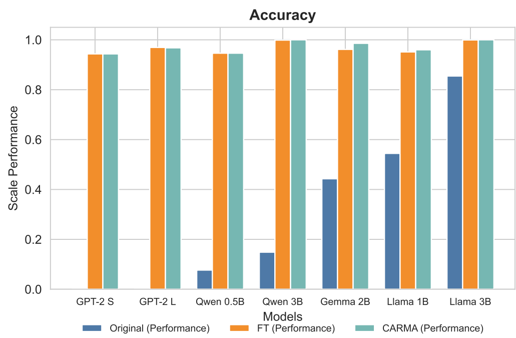


Figure 4: Task performance in SC across GPT2 (S, L), Gemma-2B, Llama (1B, 3B) and Qwen (0.5B, 3B).

5.3 Impact of CARMA on Performance

Fig. 3 and 4 show the performance of original, FT, and CARMA accuracies across tasks. CARMA demonstrates significant improvements over original models across tasks. For example, in IDM, GPT2-L achieves 150% improvement, and Llama-3B shows an 89.6% increase, while in SC, Gemma-2B demonstrates 122.5% improvement over Original baselines.

Task-specific patterns emerge when comparing models. For example, in IDM, CARMA outperforms FT, with Llama-3B showing a +5% gain and GPT2-L improving by 1.7%. In SC, CARMA maintains comparable performance to FT while enhancing robustness, suggesting it preserves learned features while strengthening compositional consistency.

CARMA enhances FT by improving representation stability and preventing feature drift, ensuring structured compositional consistency. Its benefits are most pronounced in larger models, where greater capacity supports robust representations while maintaining fine-tuned performance. This scalability highlights CARMA's effectiveness in regularising model representations and reinforcing compositional structure without disrupting learned task features, providing a reliable solution for improving compositional reasoning in LLMs.

6 Related work

Research on CG in LLMs has revealed both capabilities and limitations (Tull et al., 2024; Moisio et al., 2023; Sinha et al., 2024), though many studies lack mechanistic analysis or concrete suggestions for improvements.

Architectural modifications are a common approach to tackle CG challenges. Recent proposals include pushdown layers for recursive attention (Murty et al., 2023), Layer-wise Representation Fusion for dynamic encoder weighting (Lin et al., 2023), and specialised semantic parsing methods (Shaw et al., 2021). While effective for specific tasks, these solutions face scalability challenges due to computational overhead, specialised annotation requirements, and architectural constraints.

Regularisation methods provide alternative approaches through consistency regularisation (Yin et al., 2023b), data augmentation strategies (Ontanon et al., 2022), and attention stability mechanisms (Zhai et al., 2023). Studies show dataset complexity and example frequency variations improve compositional reasoning (Zhou et al., 2023). However, these methods face key limitations: token-level approaches lack adaptability to complex structures, augmentation shows diminishing returns on real data, and stability mechanisms prioritise training stability over compositional generalisation.

Evaluation challenges persist in CG research. Standard benchmarks like SCAN (Lake and Baroni, 2017), PCFG (Hupkes et al., 2020), and COGS (Kim and Linzen, 2020b) rely heavily on synthetic data, limiting real-world applicability. Recent frameworks like CoGnition (Li et al., 2021) and CAP (Aljaafari et al., 2024) better align with natural language phenomena, but evaluation gaps remain. Current approaches often sacrifice generalisability for task-specific performance. *CARMA* addresses these limitations through a *task-agnostic, efficient solution* that enhances CG while maintaining robust cross-task performance.

7 Conclusion

This paper presents CARMA, a method for enhancing compositional generalisation in LLMs through mutual information regularisation and layer-wise stability constraints. By addressing information fragmentation and instability across layers, CARMA improves performance robustness and stability under intervention. The method requires no architectural changes and integrates cleanly into standard fine-tuning pipelines. Future work should explore extending CARMA to tasks requiring more nuanced semantic reasoning and to multilingual contexts. Another important direction is combining CARMA with techniques that explicitly challenge generalisation, such as adversarial perturbations or structured distribution shifts, to promote the acquisition of novel compositional behaviours. Incorporating CARMA into improved, task-targeted architectures may further enhance its effectiveness. Additionally, controlled training-from-scratch studies could isolate CARMA's impact more precisely and reveal deeper insights into how it shapes compositional representations across training regimes.

Limitations

The limitations of this paper can be summed up as follows: First, our results are primarily reported for the English language. Further analysis across languages with diverse linguistic structures is left as a confirmatory future work. Second, the datasets

(WordNet and SST) lack a more comprehensive representativeness of broader linguistic phenomena. Third, our focus is predominantly on decoder-based Transformers, and the employed Transformer models may inherit potential biases ingrained from their pre-training data. Finally, while CARMA maintains inference efficiency, it introduces training-time overhead due to auxiliary losses, which should be considered when integrating the method into resource-constrained environments.

Ethical statement

This work aims to enhance language model robustness and compositional understanding through CARMA. While improving model reliability is beneficial, we acknowledge potential risks in enhancing language model capabilities. Our evaluation focuses on controlled tasks (IDM and SC) with comprehensive stability metrics to ensure responsible development and transparent reporting of model behaviour under perturbations.

References

Nura Aljaafari, Danilo S Carvalho, and André Freitas. 2024. Interpreting token compositionality in llms: A robustness analysis. *arXiv preprint arXiv:2410.12924*.

Steven Bird, Ewan Klein, and Edward Loper. 2009. *Natural language processing with Python: analyzing text with the natural language toolkit*. " O'Reilly Media, Inc.".

Xingyu Cai, Jiaji Huang, Yuchen Bian, and Kenneth Church. 2021. Isotropy in the contextual embedding space: Clusters and manifolds. In *International conference on learning representations*.

Ta-Chung Chi, Ting-Han Fan, Alexander Rudnicky, and Peter Ramadge. 2023. Transformer working memory enables regular language reasoning and natural language length extrapolation. In *Findings of the Association for Computational Linguistics: EMNLP 2023*, pages 5972–5984, Singapore. Association for Computational Linguistics.

Róbert Csordás, Kazuki Irie, and Juergen Schmidhuber. 2021. The devil is in the detail: Simple tricks improve systematic generalization of transformers. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 619–634, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.

Verna Dankers, Elia Bruni, and Dieuwke Hupkes. 2022a. The paradox of the compositionality of natural language: A neural machine translation case study. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 4154–4175, Dublin, Ireland. Association for Computational Linguistics.

Verna Dankers, Elia Bruni, and Dieuwke Hupkes. 2022b. The paradox of the compositionality of natural language: A neural machine translation case study. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 4154–4175. Association for Computational Linguistics.

Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, Aiesha Letman, Akhil Mathur, Alan Schelten, Amy Yang, Angela Fan, et al. 2024. The llama 3 herd of models. *arXiv preprint arXiv:2407.21783*.

Nouha Dziri, Ximing Lu, Melanie Sclar, Xiang Lorraine Li, Liwei Jiang, Bill Yuchen Lin, Sean Welleck, Peter West, Chandra Bhagavatula, Ronan Le Bras, et al. 2024. Faith and fate: Limits of transformers on compositionality. *Advances in Neural Information Processing Systems*, 36.

Christiane Fellbaum. 1998. Wordnet: An electronic lexical database. *MIT Press google schola*, 2:678–686.

Sheridan Feucht, David Atkinson, Byron Wallace, and David Bau. 2024. Token erasure as a footprint of implicit vocabulary items in llms. In *The 2024 Conference on Empirical Methods in Natural Language Processing*.

Linyang He, Peili Chen, Ercong Nie, Yuanning Li, and Jonathan R. Brennan. 2024. Decoding probing: Revealing internal linguistic structures in neural language models using minimal pairs. In *Proceedings of the 2024 Joint International Conference on Computational Linguistics, Language Resources and Evaluation (LREC-COLING 2024)*, pages 4488–4497, Torino, Italia. ELRA and ICCL.

Matthew Honnibal, Ines Montani, Sofie Van Landeghem, and Adriane Boyd. 2020. spacy: Industrial-strength natural language processing in python.

Dieuwke Hupkes, Verna Dankers, Mathijs Mul, and Elia Bruni. 2020. Compositionality decomposed: How do neural networks generalise? (extended abstract). In *Proceedings of the Twenty-Ninth International Joint Conference on Artificial Intelligence, IJCAI-20*, pages 5065–5069. International Joint Conferences on Artificial Intelligence Organization. Journal track.

Mete Ismayilzada, Defne Circi, Jonne Sälevä, Hale Sirin, Abdullatif Köksal, Bhuwan Dhingra, Antoine Bosselut, Lonneke van der Plas, and Duygu Ataman. 2024. Evaluating morphological compositional generalization in large language models. *arXiv preprint arXiv:2410.12656*.

Najoung Kim and Tal Linzen. 2020a. Cogs: A compositional generalization challenge based on semantic interpretation. In *Proceedings of the 2020 conference on empirical methods in natural language processing (emnlp)*, pages 9087–9105.

Najoung Kim and Tal Linzen. 2020b. COGS: A compositional generalization challenge based on semantic interpretation. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 9087–9105, Online. Association for Computational Linguistics.

Nikita Kitaev, Steven Cao, and Dan Klein. 2019. Multilingual constituency parsing with self-attention and pre-training. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 3499–3505, Florence, Italy. Association for Computational Linguistics.

Nikita Kitaev and Dan Klein. 2018. Constituency parsing with a self-attentive encoder. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 2676–2686, Melbourne, Australia. Association for Computational Linguistics.

Brenden M. Lake and Marco Baroni. 2017. Generalization without systematicity: On the compositional skills of sequence-to-sequence recurrent networks. In *International Conference on Machine Learning*.

Anna Langedijk, Hosein Mohebbi, Gabriele Sarti, Willem Zuidema, and Jaap Jumelet. 2024. DecoderLens: Layerwise interpretation of encoder-decoder transformers. In *Findings of the Association for Computational Linguistics: NAACL 2024*, pages 4764–4780, Mexico City, Mexico. Association for Computational Linguistics.

Yafu Li, Yongjing Yin, Yulong Chen, and Yue Zhang. 2021. On compositional generalization of neural machine translation. 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)*, pages 4767–4780, Online. Association for Computational Linguistics.

Zhaoyi Li, Ying Wei, and Defu Lian. 2023. Learning to substitute spans towards improving compositional generalization. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 2791–2811, Toronto, Canada. Association for Computational Linguistics.

Lei Lin, Shuangtao Li, Yafang Zheng, Biao Fu, Shan Liu, Yidong Chen, and Xiaodong Shi. 2023. Learning to compose representations of different encoder layers towards improving compositional generalization. In *Findings of the Association for Computational Linguistics: EMNLP 2023*, pages 1599–1614, Singapore. Association for Computational Linguistics.

Steven Loria et al. textblob documentation. *Release 0.18.0*.

George A. Miller. 1994. WordNet: A lexical database for English. In *Human Language Technology: Proceedings of a Workshop held at Plainsboro, New Jersey, March 8-11, 1994*.

Anssi Moisio, Mathias Creutz, and Mikko Kurimo. 2023. Evaluating morphological generalisation in machine translation by distribution-based compositionality assessment. In *Proceedings of the 24th Nordic Conference on Computational Linguistics (NoDaLiDa)*, pages 738–751, Tórshavn, Faroe Islands. University of Tartu Library.

Shikhar Murty, Pratyusha Sharma, Jacob Andreas, and Christopher Manning. 2023. Pushdown layers: Encoding recursive structure in transformer language models. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 3233–3247, Singapore. Association for Computational Linguistics.

Neel Nanda and Joseph Bloom. 2022. Transformerlens. `https://github.com/TransformerLensOrg/TransformerLens`.

Santiago Ontanon, Joshua Ainslie, Zachary Fisher, and Vaclav Cvicek. 2022. Making transformers solve compositional tasks. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 3591–3607, Dublin, Ireland. Association for Computational Linguistics.

Aaron van den Oord, Yazhe Li, and Oriol Vinyals. 2018. Representation learning with contrastive predictive coding. *arXiv preprint arXiv:1807.03748*.

Barbara H. Partee. 1984. Compositionality. In Fred Landman and Frank Veltman, editors, *Varieties of Formal Semantics*, pages 281–312. Foris Publications.

Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, et al. 2019. Pytorch: An imperative style, high-performance deep learning library. *Advances in neural information processing systems*, 32.

F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay. 2011. Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, 12:2825–2830.

Jackson Petty, Sjoerd Steenkiste, Ishita Dasgupta, Fei Sha, Dan Garrette, and Tal Linzen. 2024. The impact of depth on compositional generalization in transformer language models. In *Proceedings of the 2024 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers)*, pages 7232–7245.

Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, Ilya Sutskever, et al. 2019. Language models are unsupervised multitask learners. *OpenAI blog*, 1(8):9.

Parikshit Ram, Tim Klinger, and Alexander Gray. 2024. What makes models compositional? a theoretical view. In *Proceedings of the Thirty-Third International Joint Conference on Artificial Intelligence (IJCAI-24)*, pages 4824–4832. International Joint Conferences on Artificial Intelligence Organization.

Rahul Ramesh, Ekdeep Singh Lubana, Mikail Khona, Robert P. Dick, and Hidenori Tanaka. 2024. Compositional capabilities of autoregressive transformers: A study on synthetic, interpretable tasks. In *Forty-first International Conference on Machine Learning*.

Mehdi Sajjadi, Mehran Javanmardi, and Tolga Tasdizen. 2016. Regularization with stochastic transformations and perturbations for deep semi-supervised learning. *Advances in neural information processing systems*, 29.

Peter Shaw, Ming-Wei Chang, Panupong Pasupat, and Kristina Toutanova. 2021. Compositional generalization and natural language variation: Can a semantic parsing approach handle both? 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)*, pages 922–938, Online. Association for Computational Linguistics.

Ankur Sikarwar, Arkil Patel, and Navin Goyal. 2022. When can transformers ground and compose: Insights from compositional generalization benchmarks. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pages 648–669, Abu Dhabi, United Arab Emirates. Association for Computational Linguistics.

Sania Sinha, Tanawan Premsri, and Parisa Kordjamshidi. 2024. A survey on compositional learning of ai models: Theoretical and experimetnal practices. *arXiv preprint arXiv:2406.08787*.

Richard Socher, Alex Perelygin, Jean Wu, Jason Chuang, Christopher D Manning, Andrew Y Ng, and Christopher Potts. 2013. Recursive deep models for semantic compositionality over a sentiment treebank. In *Proceedings of the 2013 conference on empirical methods in natural language processing*, pages 1631–1642.

Gemma Team, Morgane Riviere, Shreya Pathak, Pier Giuseppe Sessa, Cassidy Hardin, Surya Bhupatiraju, Léonard Hussenot, Thomas Mesnard, Bobak Shahriari, Alexandre Ramé, et al. 2024. Gemma 2: Improving open language models at a practical size. *arXiv preprint arXiv:2408.00118*.

Sean Tull, Robin Lorenz, Stephen Clark, Ilyas Khan, and Bob Coecke. 2024. Towards compositional interpretability for xai. *arXiv preprint arXiv:2406.17583*.

Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Rémi Louf, Morgan Funtowicz, Joe Davison, Sam Shleifer, Patrick von Platen, Clara Ma, Yacine Jernite, Julien Plu, Canwen Xu, Teven Le Scao, Sylvain Gugger, Mariama Drame, Quentin Lhoest, and Alexander Rush. 2019. Huggingface's transformers: State-of-the-art natural language processing. *arXiv preprint arXiv:1910.03771*.

Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Rémi Louf, Morgan Funtowicz, Joe Davison, Sam Shleifer, Patrick von Platen, Clara Ma, Yacine Jernite, Julien Plu, Canwen Xu, Teven Le Scao, Sylvain Gugger, Mariama Drame, Quentin Lhoest, and Alexander M. Rush. 2020. Transformers: State-of-the-art natural language processing. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 38–45, Online. Association for Computational Linguistics.

An Yang, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chengyuan Li, Dayiheng Liu, Fei Huang, Haoran Wei, et al. 2024a. Qwen2.5 technical report. *arXiv preprint arXiv:2412.15115*.

Haoran Yang, Hongyuan Lu, Wai Lam, and Deng Cai. 2024b. Exploring compositional generalization of large language models. In *Proceedings of the 2024 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 4: Student Research Workshop)*, pages 16–24.

Yongjing Yin, Jiali Zeng, Yafu Li, Fandong Meng, Jie Zhou, and Yue Zhang. 2023a. Consistency regularization training for compositional generalization. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1294–1308.

Yongjing Yin, Jiali Zeng, Yafu Li, Fandong Meng, Jie Zhou, and Yue Zhang. 2023b. Consistency regularization training for compositional generalization. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1294–1308, Toronto, Canada. Association for Computational Linguistics.

Shuangfei Zhai, Tatiana Likhomanenko, Etai Littwin, Dan Busbridge, Jason Ramapuram, Yizhe Zhang, Jiatao Gu, and Joshua M Susskind. 2023. Stabilizing transformer training by preventing attention entropy collapse. In *International Conference on Machine Learning*, pages 40770–40803. PMLR.

Min Zhang, Jianfeng He, Shuo Lei, Murong Yue, Linhan Wang, and Chang-Tien Lu. 2024. Can llm find the green circle? investigation and human-guided tool manipulation for compositional generalization. In *ICASSP 2024-2024 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 11996–12000. IEEE.

Tianqi Zhong, Zhaoyi Li, Quan Wang, Linqi Song, Ying Wei, Defu Lian, and Zhendong Mao. 2024. Benchmarking and improving compositional generalization of multi-aspect controllable text generation. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 6486–6517. Association for Computational Linguistics.

Xiang Zhou, Yichen Jiang, and Mohit Bansal. 2023. Data factors for better compositional generalization. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 14549–14566, Singapore. Association for Computational Linguistics.

A Task Selection and Compositionality Considerations

To assess compositional generalisation and the benefits of CARMA, we targeted tasks that involve systematic meaning construction and sensitivity to structural modifications. To that end, we opted to employ Inverse Dictionary Modelling (IDM) and Sentiment Classification (SC) as proxies for different dimensions of compositionality, capturing both structured composition and hierarchical generalisation.

IDM requires models to generate a single-word representation from a natural language definition, mapping from the composition of input constituents (individual concept components) to a specific term. On the other hand, SC maps meaning to a sentiment label, aggregating local meaning elements into a global interpretation. While IDM focuses on explicit compositional mapping, SC evaluates distributed composition, where sentiment is shaped by multiple interacting components.

Both tasks assess several aspects of compositionality (Figure 5), namely systematicity (structured meaning formation), substitutivity (semantic preservation under transformation), and resistance to over-generalisation (ensuring bounded semantic deviation). Further, they evaluate robustness, testing whether models can maintain correctness and consistency under internal and input-lexical perturbations. IDM and SC provide a comprehensive test of compositional generalisation across structured and distributed representations.

B Detailed Experimental Configuration

B.1 Task Formalisation

This paper evaluates the effectiveness of CARMA in enhancing the compositional generalisation of large language models (LLMs) through two tasks.

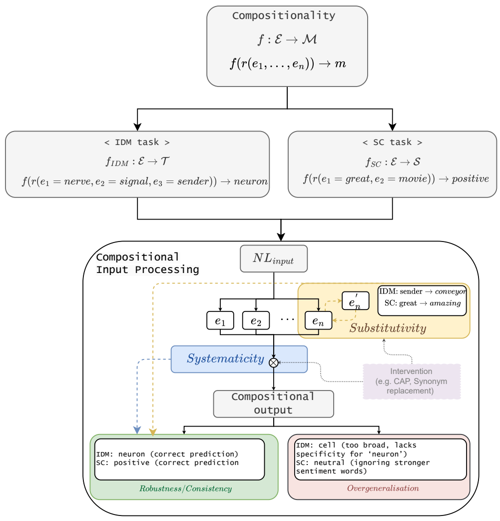


Figure 5: Illustration of compositional generalisation in Inverse Dictionary Modelling (IDM) and Sentiment Classification (SC). The figure highlights key compositional properties: systematicity ensures coherent meaning construction, substitutivity maintains meaning under lexical variations, robustness preserves intended outputs under perturbations, and over-generalisation leads to overly broad or semantically weak predictions (e.g., neuron misclassified as cell or positive reduced to neutral).

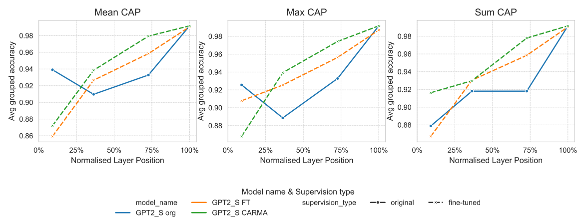
These tasks were selected based on their focus on input token structure and compositional semantics, utilising next-token prediction with single-token outputs. Formal definitions for each task are presented below.

Inverse Definition Modelling (IDM). This task requires the model to predict a definiendum D, given its corresponding definition definition in natural language. Formally, the definition is represented as a sequence of tokens, $\text{definition} = \{\text{tok}_1, \text{tok}_2, \ldots, \text{tok}_n\}$, and the model seeks to produce D such that:

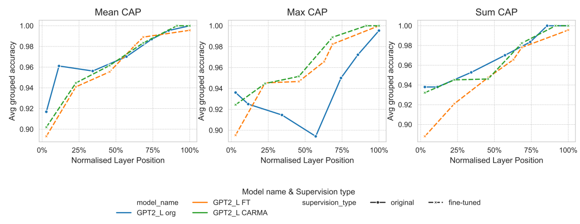
$$D = \arg\max_{t \in \mathcal{V}} P(d \mid \text{definition}), \tag{9}$$

where $\mathcal{V}$ denotes the model's vocabulary, and d represents a potential definiendum. Predictions are deemed correct only if they exactly match the target output.

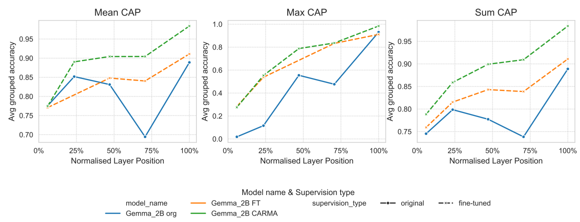
Sentiment classification (SC). This task involves assigning a sentiment label to a given sentence containing sentiment cues and potential modifiers. The model processes the input sentence, represented as a sequence of tokens $\text{sentence} =$



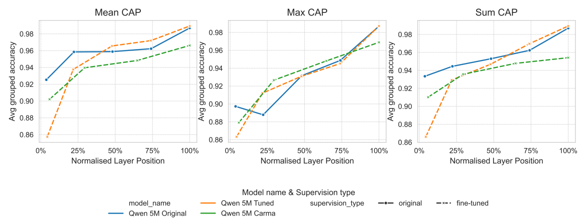
(a) GPT2-S



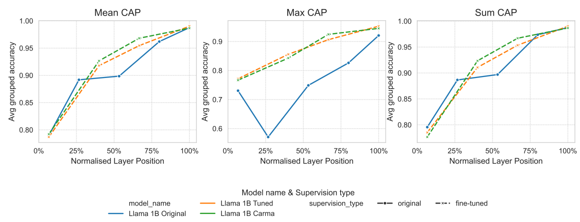
(b) GPT2-L



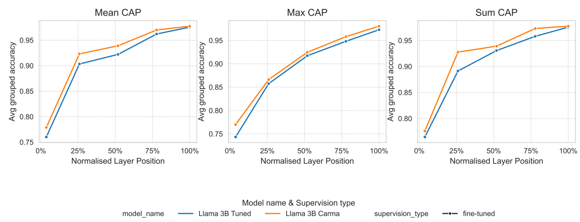
(c) Gemma-2B



(d) Qwen-0.5B



(e) Llama-1B



(f) Llama-3B

Figure 6: IDM Performance Across Models Under CAP

$\{\text{tok}_1, \text{tok}_2, \ldots, \text{tok}_n\}$, and produces an output label from a predefined set of sentiment classes $\mathcal{A}$ (i.e., *positive*, *negative*, *neutral*). Formally, the task is defined as:

$$\text{label} = \arg\max_{\ell \in \mathcal{L}} P(\ell \mid \text{sentence}), \tag{10}$$

where $P(\ell \mid \text{sentence})$ is the probability of the sentiment label ℓ given the sentence. The model's performance is evaluated based on its ability to correctly predict the sentiment, accounting for compositional nuances such as modifiers and contrasts.

B.2 Datasets specification and pre-processing

For IDM, the training and test datasets were derived from WordNet (Fellbaum, 1998), a widely used lexical database of the English language. WordNet comprises over 117,000 synsets, each representing a distinct concept and annotated with semantic relationships such as hypernyms, synonyms, and definitions. To ensure consistency and improve data quality, standard preprocessing techniques were applied, including the removal of special characters, punctuation, extra spaces, and parenthesised content where necessary. The dataset focuses on general-purpose vocabulary rather than specialised domains or demographic groups. The dataset was initially split into an 80-20 ratio, with 80% allocated for training. The remaining 20% was further divided equally into validation and test sets.

The SC dataset was derived from the Stanford Sentiment Treebank (SST) (Socher et al., 2013), a corpus of English movie reviews annotated for analysis of the compositional effects of sentiment inference and was released under Apache License, Version 2.0. SST includes fine-grained sentiment labels at both the phrase and sentence levels, making it a standard benchmark for evaluating sentiment classification models. The original dataset splits provided by the authors were maintained to ensure consistency in training, validation, and testing. For SST labels, sentiment scores were categorised as follows: values equal to or greater than 0.6 were classified as positive, scores between 0 and 0.6 were considered neutral, and scores below zero were assigned as negative. The final test dataset sizes for each task are presented in Table 2.

Dataset	Train size	validation Size	Test Size
WordNet	9563	1154	1231
SST	8544	1101	2210

Table 2: Train, validation, and test set sizes for WordNet and SST datasets used in this paper.

B.3 Model training and fine-tuning settings

Table 3 summarises the key characteristics of the models evaluated in this study. All models were obtained from Hugging Face (Wolf et al., 2019) under their respective licenses: GPT-2 (Modified MIT), Llama 3.2 (Meta Llama 3 Community), Qwen 2.5 (Apache 2.0), and Gemma-2B (Gemma Terms of Use). While all models were pre-trained on English data, LLama and Qwen models provide additional multilingual capabilities, namely English, German, French, Italian, Portuguese, Hindi, Spanish, and Thai for LLama, and over 10 languages, including Chinese, English, French, Spanish, Portuguese, Russian, Arabic, Japanese, Korean, Vietnamese, Thai, and Indonesian for Qwen. The models employ the following tokenisation approaches: GPT-2, Byte Pair Encoding (BPE) with a 50,257-token vocabulary, optimised primarily for English, Llama 3.2 uses SentencePiece-based BPE, combining 100K tokens from Tiktoken3 with 28K additional tokens to enhance multilingual performance, Qwen 2.5 employs Byte-level BPE, utilising a 151,643-token vocabulary designed for multilingual processing, Gemma-2B has a SentencePiece tokeniser leveraging a 256,000-token vocabulary, making it highly effective for English-based tasks. Each model was fine-tuned on its respective downstream task following a systematic hyperparameter search to identify optimal configurations. Prior to fine-tuning, prompt engineering was conducted to determine well-performing prompts tailored to each task, ensuring alignment with task-specific requirements and enhancing the models' ability to generate accurate and contextually relevant outputs. The hyperparameter search explored key factors, including weights for stability regularisation, mutual information (MI) regularisation, and the overall CARMA weight (Equation 7), as well as the specific layers to which these losses were applied.

For training parameters, the following batch sizes were set in the IDM task: 16 for the Gemma-2B and GPT models, 32 for the Qwen-3B and Llama models, and 64 for the Qwen-0.5B model. For SC, the batch sizes were 16 for the GPT models, Gemma-2B and Llama-3B; 32 for Llama-1B and Qwen-3B; and 64 for Qwen-0.5B. For the number of training epochs, in the IDM, the Gemma and GPT models were trained for two epochs, while all other models were trained for three epochs, whereas all models were trained for two epochs, except Gemma-2B and LLama-1B, which were trained for three epochs for the SC task. The stopping layers for IDM and CARMA were configured as follows: GPT2-S at layer 3, GPT2-L at layer 8, Gemma-2B at layer 10, Llama-1B at layer 7, Llama-3B at layers 8 (stability) and 12 (MI), Qwen-0.5B at layer 5, and Qwen-3B at layer 10. The SC, the ending layers, 4 for GPT2-S, 12 for GPT2-L, 10 for Gemma-2B, 7, for LLama 1B, 8, for LLama 3B, 5 for Qwen-0.5B and 7 for Qwen-3B. For CARMA weight, optimal values varied by model size: 0.4 and 0.5 were most effective for larger models. We hypothesise that CARMA regularisation exhibits a weaker effect when lower weights are applied, particularly in larger architectures where stronger constraints are needed to stabilise compositional representations. In IDM, GPT2-L and Gemma performed best with a weight of 0.3, GPT2-S with 0.2, Llama-1B with 0.4, and Llama-3B with 0.5. Qwen models used 0.5 and 0.4 for the 0.5B and 3B variants, respectively. For SC Carma weight, it was 0.4 for Qwen-0.5B and GPT models, 0.5 for LLama-3B and Qwen-3B, and 0.3 for the rest. For the ending layer, it was 4 for GPT2-S, 12 for GPT2-L, 10 for Gemma-2B, 7 for LLama-1B, 8 for LLama-3B, 5 for Qwen-0.5B and 7 for Qwen-3B.

Model	Parameters	Layers	D_{model}	Heads	Activation	MLP Dimension
GPT-2 Small	85M	12	768	12	GELU	3072
GPT-2 Large	708M	36	1280	20	GELU	5120
Gemma-2B	2B	32	4096	16	GELU	8192
LLaMA3.2 1B	1.1B	16	2048	32	SiLU	8192
LLaMA3.2 3B	3.2B	28	3072	24	SiLU	8192
Qwen2.5-0.5B	391M	24	896	14	SiLU	4864
Qwen2.5-3B	3.0B	36	2048	16	SiLU	11008

Table 3: Summary of model architectures. **Parameters**: total number of trainable parameters; **Layers**: total number of transformer layers; $\mathbf{D_{model}}$: size of word embeddings and hidden states; **Heads**: number of self-attention heads; **Activation**: activation function used in feedforward layers; **MLP Dimension**: dimensionality of the feedforward network.

B.4 Evaluation Metrics

This section details the evaluation metrics used in the study, including accuracy, synonym consistency, and performance stability.

Accuracy is used as a primary measure of model performance and is defined as:

$$\text{Accuracy} = \frac{TP + TN}{TP + TN + FP + FN}, \quad (11)$$

where TP (true positives) and TN (true negatives) denote correctly classified instances, while FP (false positives) and FN (false negatives) represent misclassified instances.

Synonym Consistency (ConsistSyn) quantifies a model's ability to maintain correct predictions after synonym replacement. It is computed as:

$$\text{ConsistSyn} = \frac{|\text{Correct After Replacement}|}{|\text{Correct Before Replacement}|} \times 100, \quad (12)$$

where Correct After Replacement refers to the number of correct predictions following synonym substitution, and Correct Before Replacement denotes the number of correct predictions before substitution. The reported results are the averaged ConsistSyn across ($N \geq 5$) runs.

Coefficient of Variation (CV) measures the stability of model performance across multiple runs, with lower values indicating greater consistency. It is defined as:

$$\text{CV} = \frac{\sigma}{\mu}, \quad (13)$$

where σ represents the standard deviation of model performance across runs, and μ denotes the mean performance.

Normalised Improvement (NI) evaluates the relative gain in consistency introduced by a model over a baseline model. It is calculated as:

$$\text{NI} = \frac{\text{ConsistSyn}_{\text{CARMA}} - \text{ConsistSyn}_{\text{baseline}}}{\text{ConsistSyn}_{\text{baseline}}} \times 100. \quad (14)$$

This metric captures the percentage improvement in synonym consistency due to a model variant compared to the baseline model.

B.5 Experimental setup

Experiments were conducted using NVIDIA RTX A6000 and A100 GPUs. The method was developed in Python (v3.10.15) with Transformers (v4.44.2) (Wolf et al., 2020), PyTorch (v2.4.1) (Paszke et al., 2019), and Transformer-lens (v2.8.1) (Nanda and Bloom, 2022). Preprocessing tasks, including tokenisation and tagging, used NLTK (v3.9.1) (Bird et al., 2009), spaCy (v3.7.2) (Honnibal et al., 2020), and TextBlob (v0.18.0) (Loria et al.), with Scikit-learn (v1.5.1) (Pedregosa et al., 2011) for evaluation. Models use 500 warm-up steps and a 0.006 learning rate.

C Comprehensive Explanation of Evaluation Interventions

C.1 Constituent-Aware Pooling (CAP) Formalisation

Constituent-Aware Pooling (CAP) Formalisation is a method proposed in (Aljaafari et al., 2024) to systematically assess compositional generalisation via aggregating token-level activations into higher-level semantic representation. Below is a detailed explanation and formalisation of CAP.

Overview. CAP aggregates model activations at any chosen constituency level (e.g. tokens to words), enabling the analysis of compositional dependencies. The key steps involved are:

- **Input Representations:** For a given input sequence $X = [x_1, x_2, \ldots, x_n]$, the model produces inner states $H = [h_1, h_2, \ldots, h_n]$ at a specific layer.
- **Grouping Constituents:** Using syntactic parsers such as Benepar (Kitaev et al., 2019; Kitaev and Klein, 2018), or by inversing the model tokeniser function, the sequence is segmented into constituents $C = [c_1, c_2, \ldots, c_m]$, where each c_i represents a phrase or syntactic unit. For the experiments presented in the paper, tokens were grouped into words to form the smallest linguistic units.
- **Pooling Operations:** For each constituent c_i, the corresponding activations $\{h_j | x_j \in c_i\}$ are aggregated into a single representation r_i using a pooling function:

  $$r_i = \alpha(\{h_j | x_j \in c_i\})$$

  CAP supports three pooling functions:

  - **Maximum pooling:** Selects the highest activation values as:

    $$\alpha(\{h_j | x_j \in c_i\}) = \max(\{h_j | x_j \in c_i\}),$$

  - **Mean pooling:** Computes the average of activation values as:

    $$\alpha(\{h_j | x_j \in c_i\}) = \frac{1}{|c_i|} \sum_{j \in c_i} \{h_j | x_j \in c_i\},$$

– **Sum pooling:** Accumulates activation values as:

$$\alpha(\{h_j|x_j \in c_i\}) = \sum_{j \in c_i} \{h_j|x_j \in c_i\}.$$

- **Updating Representations:** The pooled representations $R = [r_1, r_2, \ldots, r_m]$ replace the original activations H for further processing.

Evaluation. The impact of CAP is evaluated by comparing task-specific performance metrics (e.g., accuracy, F1 score) of models before and after CAP is applied. This allows for a direct assessment of how CAP affects compositionality and task performance. This paper utilises the word-level CAP, pooling related token representation to their corresponding words.

C.2 Synonym Replacement

A multi-step approach was adopted to ensure reliable synonym replacements. First, preprocessing was applied to filter out words that were unlikely to produce meaningful replacements. Specifically, words belonging to NLTK's predefined stopwords list or shorter than two characters were excluded from consideration. The remaining words were tagged with their part-of-speech (POS) using spaCy's (Honnibal et al., 2020) POS tagger. Additionally, the sentiment of each word was determined using TextBlob (Loria et al.) to ensure that replacements preserved the semantic tone of the original text. Next, a synonym vocabulary was constructed using words extracted from spaCy's *en_core_web_md* language model. This vocabulary was filtered to include only alphabetic common words with high probability scores (greater than -15 in our case), as determined by spaCy's word frequency data, while stopwords and rare terms were excluded. This step ensured that the vocabulary consisted of meaningful and contextually appropriate words for replacement. For each target word, a list of synonym candidates was generated by iterating over the constructed vocabulary. The top n candidates were selected based on their semantic similarity to the original word, measured using spaCy's word vectors. Synonyms with high similarity scores and alignment in POS were prioritised to maintain grammatical and contextual coherence in the text.

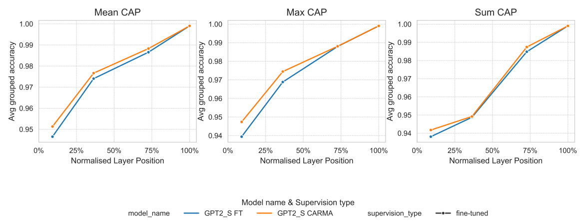
Model	Ver.	Task	Int.	CS	CV
GPT2-S	CARMA	IDM	25%	49.17	0.025
	FT	IDM	25%	50.89	**0.017**
	Org	IDM	25%	**52.46**	0.044
	CARMA	IDM	40%	35.90	**0.0542**
	FT	IDM	40%	37.16	0.0628
	Org	IDM	40%	**37.20**	0.1223
GPT2-L	CARMA	IDM	25%	56.31	**0.0164**
	FT	IDM	25%	**56.95**	0.0311
	Org	IDM	25%	51.10	0.1175
	CARMA	IDM	40%	43.56	0.0485
	FT	IDM	40%	**43.97**	**0.0459**
	Org	IDM	40%	34.68	0.0895
Gemma-2B	CARMA	IDM	25%	56.70	**0.023**
	FT	IDM	25%	**57.42**	0.030
	Org	IDM	25%	49.47	0.031
	CARMA	IDM	40%	0.4236	**0.0174**
	FT	IDM	40%	**0.4498**	0.0249
	Org	IDM	40%	0.3576	0.0480
Llama-1B	CARMA	IDM	25%	**58.40**	0.0400
	FT	IDM	25%	57.86	**0.0385**
	Org	IDM	25%	47.55	0.0503
	CARMA	IDM	40%	**47.07**	0.0476
	FT	IDM	40%	46.75	0.0455
	Org	IDM	40%	33.49	**0.0391**
Qwen-0.5B	CARMA	IDM	25%	**56.98**	0.0286
	FT	IDM	25%	54.57	**0.0191**
	Org	IDM	25%	46.84	0.0684
	CARMA	IDM	40%	**40.55**	**0.0397**
	FT	IDM	40%	39.69	0.0491
	Org	IDM	40%	32.98	0.0938
Qwen-3B	CARMA	IDM	25%	**62.00**	**0.0225**
	FT	IDM	25%	61.79	0.0279
	Org	IDM	25%	49.37	0.0441
	CARMA	IDM	40%	45.05	**0.0400**
	FT	IDM	40%	**45.74**	0.0551
	Org	IDM	40%	31.95	0.0688
Llama-3B	CARMA	IDM	25%	**62.86**	**0.015**
	FT	IDM	25%	62.22	0.029
	Org	IDM	25%	52.47	0.035
	CARMA	IDM	40%	**49.05**	0.0297
	FT	IDM	40%	48.31	**0.0191**
	Org	IDM	40%	36.95	0.0458

Table 4: Model performance (25% and 40% synonym intervention) on the IDM task. **Ver.**: Version; **Int.**: Intervention rate; **CS**: ConsistSyn (%); **CV**: Coefficient of Variation. **Best values in bold.**

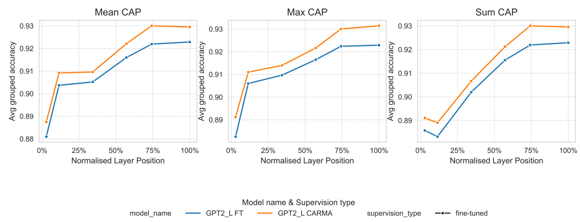
D InfoNCE for Mutual Information Estimation

Mutual information (MI) quantifies the shared information between two variables X and Y. CARMA leverages MI maximisation to capture dependencies between tokens effectively, thereby enhancing compositional generalisation in LLMs. Specifically, CARMA uses MI, denoted as $I(X;Y)$, to reinforce token-level interactions critical for compositionality. However, direct computation of MI is challenging in practice.

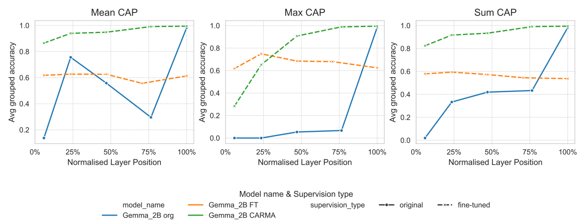
To address this challenge, a variant of InfoNCE is employed to estimate MI and approximate these



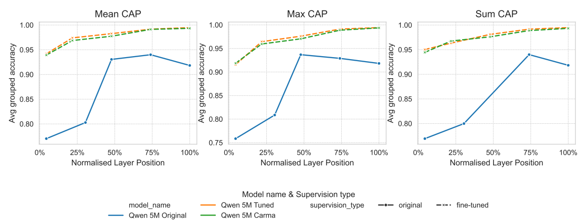
(a) GPT2-S



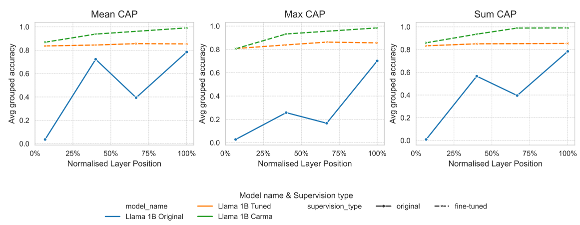
(b) GPT2-L



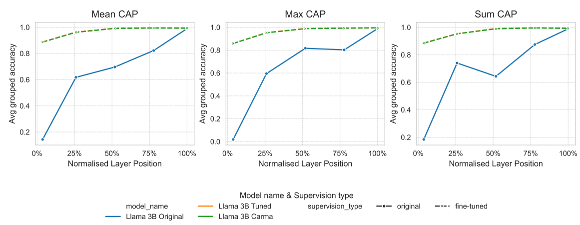
(c) Gemma-2B



(d) Qwen-0.5B



(e) Llama-1B



(f) Llama-3B

Figure 7: SC Performance Across Models Under CAP

dependencies efficiently. Given an anchor token hidden state h_i, we construct a corresponding positive set $\mathbf{H}$, which contains tokens hidden states semantically or syntactically related to h_i. Additionally, we define $\mathcal{N}$ as the set of negative examples consisting of unrelated tokens hidden states.

The InfoNCE objective provides a practical lower bound on $I(X;Y)$ (Oord et al., 2018), as follows:

$$I(X;Y) \geq \mathbb{E}\left[\log \frac{\sum_{h_j \in \mathbf{H}} f(h_i, h_j)}{\sum_{h_j \in \mathbf{H}} f(h_i, h_j) + \sum_{h_k \in \mathcal{N}} f(h_i, h_k)}\right], \tag{15}$$

where $f(h_i, h_j) = \exp(\text{sim}(h_i, h_j)/\tau)$ is a scaled similarity function, and τ is a temperature parameter. This adaptation of InfoNCE introduces token-specific interactions within the layer-wise structure of LLMs, ensuring that dependencies are captured across layers. By maximising mutual information, CARMA aligns the optimisation direction to enhance compositional structures.

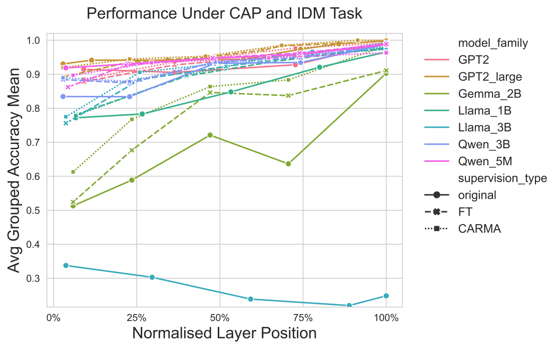
To extend this approach across layers, the final CARMA MI loss is computed as:

$$\begin{aligned}\mathcal{L}_{\text{MI}} = -\frac{1}{N}\sum_{i=1}^{N}\Bigg(\log \sum_{\substack{h_j \in \mathbf{H} \\ j \neq i}} \exp\left(\frac{\text{sim}(h_i, h_j)}{\tau}\right) \\ - \log\Bigg(\sum_{\substack{h_j \in \mathbf{H} \\ j \neq i}} \exp\left(\frac{\text{sim}(h_i, h_j)}{\tau}\right) \\ + \sum_{h_k \in \mathcal{N}} \exp\left(\frac{\text{sim}(h_i, h_k)}{\tau}\right)\Bigg)\Bigg),\end{aligned} \tag{16}$$

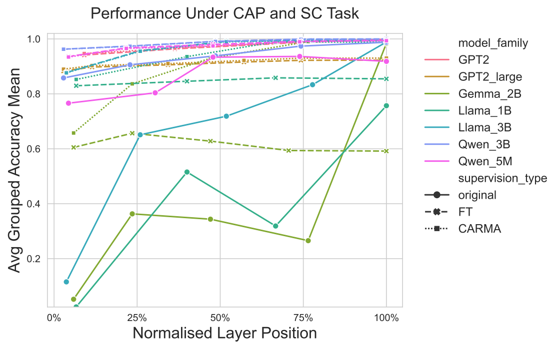
where h_i is the anchor token, $h_j \in \mathbf{H}$ are positive examples related to h_i, $h_k \in \mathcal{N}$ are negative examples, N is the number of anchors, and $\text{sim}(h_i, h_j)$ is a similarity function. The negative sign ensures that MI is maximised during optimisation. Without this negative sign, the objective would incorrectly minimise MI, thereby hindering CG enhancement.

E Extended results

Figures 6, 7 and Tables 4 and 5 provide additional results for models' performance comparison under CAP and synonym interventions. Figure 8 shows an overall models performance under cap for all models. CARMA models show a clear advantage



(a) IDM Task



(b) SC Task

Figure 8: Layer-wise performance under CAP interventions on the IDM (left) and SC (right) tasks. Results are averaged over three pooling strategies (Mean, Max, Sum) and reported for Original, Fine-Tuned (FT), and CARMA (FT + CARMA) models. Layer indices are normalised to support comparison across model sizes. CARMA improves robustness and systematicity in early-to-mid layers for both tasks, with diminishing differences in deeper layers.

Model	Ver.	Task	Int.	CS	CV
GPT2-S	CARMA	SC	25%	89.03	0.8903
	FT	SC	25%	**89.54**	**0.8954**
	CARMA	SC	40%	84.95	**0.0095**
	FT	SC	40%	**85.07**	0.0098
GPT2-L	CARMA	SC	25%	**88.58**	**0.0065**
	FT	SC	25%	88.04	0.0082
	CARMA	SC	40%	**84.61**	**0.0072**
	FT	SC	40%	84.04	0.0073
Gemma-2B	CARMA	SC	25%	**84.81**	**0.0069**
	FT	SC	25%	81.67	0.0088
	Org	SC	25%	68.14	0.0076
	CARMA	SC	40%	**81.48**	0.0102
	FT	SC	40%	74.29	**0.0073**
	Org	SC	40%	76.06	0.0136
Llama-1B	CARMA	SC	25%	74.03	0.0069
	FT	SC	25%	**75.69**	**0.0044**
	Org	SC	25%	2.65	0.1239
	CARMA	SC	40%	71.43	**0.0065**
	FT	SC	40%	**74.31**	0.0102
	Org	SC	40%	1.73	0.2245
Qwen-0.5B	CARMA	SC	25%	89.66	**0.0037**
	FT	SC	25%	**89.83**	0.0085
	Org	SC	25%	59.12	0.0691
	CARMA	SC	40%	86.03	0.0084
	FT	SC	40%	**86.31**	**0.0046**
	Org	SC	40%	55.27	0.0429
Qwen-3B	CARMA	SC	25%	93.65	0.0061
	FT	SC	25%	**93.85**	**0.0039**
	Org	SC	25%	67.63	0.0227
	CARMA	SC	40%	**91.26**	**0.0050**
	FT	SC	40%	**91.26**	**0.0050**
	Org	SC	40%	64.05	0.0159
Llama-3B	CARMA	SC	25%	84.83	**0.0056**
	FT	SC	25%	**85.85**	0.0065
	Org	SC	25%	35.21	0.0136
	CARMA	SC	40%	82.89	**0.0016**
	FT	SC	40%	**83.55**	0.0067
	Org	SC	40%	32.88	0.0188

Table 5: Model performance (25% and 40% synonym intervention) on the SC task. **Ver.**: Version; **Int.**: Intervention rate; **CS**: ConsistSyn (%); **CV**: Coefficient of Variation. **Best values in bold.**

over all models and tasks. However, the gain is clearer in the IDM case, where more intricate features and compositionality generalisation are required. It is also observed that the performance of the FT and CARMA models demonstrates similar curves or trends. Given this observation, we argue that CARMA's improvements stem from its learning objectives, which align closely with cross-entropy loss while explicitly addressing intermediate representation stability. The observed improvements are moderate in some cases, particularly for SC tasks. This behaviour is expected due to the limited size of the fine-tuning datasets compared to the original pretraining data used for these models. Nevertheless, larger models, such as Llama-3B and Gemma-2B, exhibit more substantial improvements with CARMA, demonstrating its scalability with model capacity.

E.1 Training Runtime and Overhead

We report wall-clock training times (in minutes) for each model under standard fine-tuning (FT) and with CARMA. Overhead is computed as the relative increase in runtime caused by the additional mutual information and stability losses. All experiments for models were conducted on a single GPU under identical batch size, optimiser settings, and hardware configuration.

CARMA introduces non-trivial training-time overhead due to auxiliary objectives, particularly for smaller models or longer sequences. However, inference costs remain unchanged, and no architectural modifications are required. We observe moderate to high training slowdowns (e.g., ×2.2–×2.9 for LLaMA-3B), with variance across models due

Model	FT (min)	CARMA (min)	Overhead ratio
GPT2 Small	1.9	3.48	$\times$1.8
GPT2 Large	6.7	8.9	$\times$1.3
Llama 3.2–1B	2.78	6.96	$\times$2.5
Llama 3.2–3B	7.55	20.2	$\times$2.2
Qwen 2.5–0.5B	2.10	4.75	$\times$2.3
Qwen 2.5–3B	2.98	5.01	$\times$1.7

Table 6: Wall-clock IDM training time and overhead introduced by CARMA. All runs use a single GPU under identical batch, optimiser, and hardware settings.

Model	FT (min)	CARMA (min)	Overhead ratio
GPT-2 Small	1.12	8.51	$\times$7.6
GPT-2 Large	6.02	13.18	$\times$2.19
LLaMA 3.2–1B	2.50	9.98	$\times$4.9
LLaMA 3.2–3B	6.58	14.06	$\times$2.13
Qwen 2.5–3B	6.01	16.86	$\times$2.8

Table 7: Wall-clock SC training time and overhead introduced by CARMA. All runs use a single GPU under identical batch, optimiser, and hardware settings.

to token length and loss computation. Optimising runtime for mutual information and stability estimation is an important direction for future efficiency improvements.