LOGOS: Neural Language Modeling via a Graph-Based Symbolic Lexical Knowledge Base

Anonymous ACL submission

Abstract

Addressing the interpretability and scaling bottlenecks of modern LLMs, we introduce LOGOS, a neuro-symbolic framework that replaces linear sequence modeling with a global Lemma-Merged Dependency Graph. By collapsing texts into a unified symbolic manifold, LOGOS encodes semantic relationships as explicit topological edges rather than implicit probabilities. LOGOS features: (1) topological compression, which exploits graph connectivity to circumvent the quadratic cost of sequence-level attention; and (2) Stochastic Multi-Mask Supervision, a protocol that compels the reconstruction of multi-hop relational dependencies. Evaluations on PTB, WikiText-2, and WikiText-103 demonstrate that LOGOS achieves competitive intrinsic performance with significantly fewer parameters than autoregressive baselines. Beyond efficiency, this explicit structural grounding provides a verifiable substrate for future research into aligned, hallucination-resistant AI systems.

1 Introduction

The landscape of natural language processing has been fundamentally shaped by Masked Language Modeling (MLM), which provides a robust framework for learning contextualized representations through self-supervised objectives (Devlin et al., 2019; Liu et al., 2019). However, as the field has progressed into the era of Large Language Models (LLMs), the prevailing paradigm has shifted toward massive parameter scaling and dense, sub-symbolic architectures (Touvron et al., 2023; OpenAI et al., 2023; Li et al., 2025; Behrouz et al., 2025). Despite their empirical success, these models face critical bottlenecks: the quadratic complexity of self-attention limits long-context reasoning (Gu and Dao, 2024; Peng et al., 2025), while the "black-box" nature of their representations complicates efforts in verifiable AI safety and factual grounding (Ji et al., 2023; Saparov et al., 2023).

Dependency structures offer a rigorous means of encoding syntactic and semantic relationships (Gildea and Jurafsky, 2002; Levy and Goldberg, 2014; Tai et al., 2015; Marcheggiani and Titov, 2017). While tree-structured and graph-based models have excelled in isolated tasks such as semantic role labeling and machine translation (Shen et al., 2019; Zhang et al., 2019; Chai et al., 2025), they are predominantly limited to sentence-level processing. Standard MLM objectives treat tokens as discrete units within linear sequences, neglecting the corpus-level symbolic connections and "small-world" properties inherent in human languages.

In this work, we propose LOGOS (Neural **L**anguage m**O**deling via a **G**raph-based symb**O**lic lexical knowledge ba**S**e), a neuro-symbolic architecture that unifies Graph Neural Networks (GNN) with a global Lemma-Merged Dependency Graph. LOGOS represents the linguistic topology as a unified manifold where unique lemma-typed forms act as singular nodes, with edges aggregating syntactic relations across the entire training corpus. This architectural shift enables *topological compression*, where long-range dependencies are resolved through efficient graph-based message passing rather than computationally expensive quadratic attention. To train this symbolic manifold, we propose Stochastic Multi-Mask Supervision (SMMS), a protocol that randomly masks subsets of nodes to compel the model to reconstruct relational subgraphs. Our main contributions are:

1. **Lemma-Merged Dependency Graph:** We introduce a global symbolic manifold that consolidates syntactic relations across sentence boundaries, enabling the topological compression of linguistic context to overcome the long-context bottleneck.

2. **Stochastic Multi-Mask Supervision:** We propose a training protocol that utilizes random word masking to supervise the recon-

struction of multi-hop dependencies, forcing the model to represent the global graph topology.

3. **Parameter-Efficient Competitive Performance:** We demonstrate that LOGOS achieves competitive intrinsic performance compared to larger transformer-based baselines on PTB and WikiText benchmarks, achieving high data efficiency by decoupling lexical storage from parametric inference.

4. **Neuro-Symbolic Interpretability:** LOGOS has potential to advance verifiable AI safety by grounding predictions in a transparent symbolic lexical knowledge base to enforce structural constraints.

2 Related Work

Graph-Based and Dependency-Aware Representations. Dependency structures have historically provided a rigorous framework for building compositional sentence representations (Gildea and Jurafsky, 2002; Socher et al., 2013; Zhao et al., 2021). Early neural approaches utilized tree-structured networks (Tai et al., 2015; Shen et al., 2019) and dependency-based graph encoders (Marcheggiani and Titov, 2017) to capture syntactic nuances that linear models often overlook. While recent work has expanded dependency information into global word-level networks (Levy and Goldberg, 2014; Vashishth et al., 2019; Peters et al., 2019), these efforts largely focus on word representation learning. In contrast, LOGOS leverages a global Lemma-Merged Dependency Graph to facilitate dynamic contextualized modeling, allowing syntactic roles to aggregate across disparate corpora to form a unified symbolic manifold for language modeling.

Graph Neural Networks in Language Modeling. Graph Neural Networks (GNNs) (Kipf and Welling, 2017; Veličković et al., 2018; Wu et al., 2021; Chai et al., 2025; Song et al., 2018; Qian et al., 2019) have been successfully applied in NLP tasks requiring explicit structural reasoning, such as semantic role labeling and machine translation (Zhang et al., 2019; Li et al., 2020). While the standard MLM paradigm remains dominated by Transformer-based architectures (Devlin et al., 2019; Liu et al., 2019), recent research adopts graph-based masking to incorporate non-sequential context (Zhang et al., 2020). LOGOS diverges from these sentence-centric models by replacing the intra-sequence attention mechanism with global message passing over a corpus-level lexical knowledge base. This shift enables LOGOS to resolve long-range dependencies through topological proximity rather than temporal sequence processing.

Neuro-Symbolic Hybridization and Interpretability. As LLMs scale to trillions of parameters (Touvron et al., 2023; OpenAI et al., 2023), their opaque, "black-box" nature has sparked a resurgence in neuro-symbolic AI (Saparov et al., 2023; Anonymous, 2025). Unlike pure DNNs that rely solely on statistical correlations, neuro-symbolic architectures aim to ground neural learning in explicit logic or symbolic structures. LOGOS contributes to this domain by grounding lemma representations in a transparent symbolic graph manifold. This transparency is critical for AI safety, alignment, and hallucination (Ji et al., 2023), providing an auditable structural substrate that current DNNs lack.

Efficiency and Long-Context Bottleneck. Recent advances in State Space Models seek to address the quadratic $O(N^2)$ complexity of Transformers (Gu and Dao, 2024). While models like Mamba provide linear scaling with sequence length, they remain fundamentally sequential in their processing of context. LOGOS offers an alternative solution via topological compression. By merging identical lemmas into single nodes, the graph acts as a small-world network where the effective diameter is significantly smaller than the linear sequence length. This allows LOGOS to achieve competitive performance while maintaining a substantially reduced parameter and memory footprint compared to current LLMs.

3 LMDG: A Graph-Based Symbolic Lexical Knowledge Base

The LOGOS framework, illustrated in Figure 1, represents a fundamental departure from linear sequence modeling by utilizing a **Lemma-Merged Dependency Graph (LMDG)** to encode linguistic information. This architecture jointly leverages: (i) hierarchical dependency structures, (ii) graph convolutional representations over a globally merged multi-sentence graph, and (iii) linguistic constraints that restrict the hypothesis space to structurally plausible predictions. The resulting model learns to encode syntactic structure while respecting universal linguistic well-formedness.

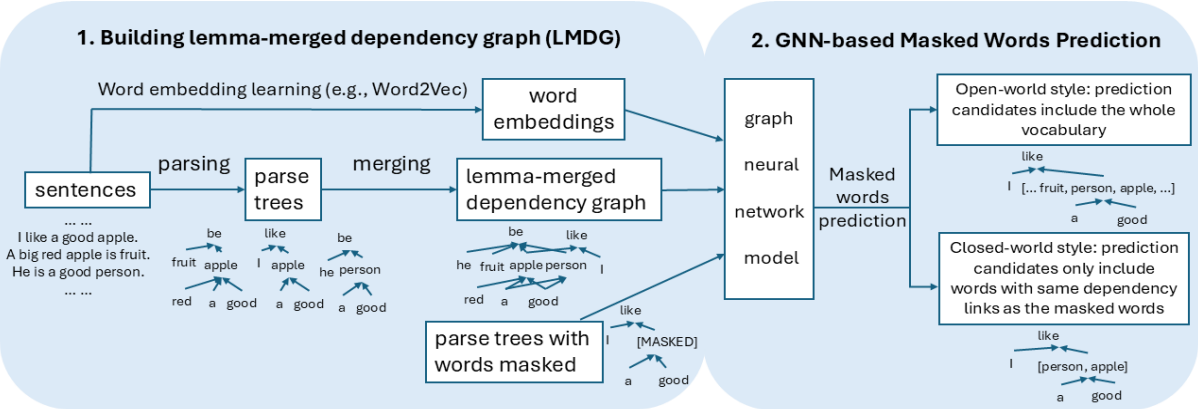

Figure 1: LOGOS framework overview

Algorithm 1 in Appendix A details the LMDG construction phase of LOGOS. Given a set of sentences $\{S_1, S_2, \ldots, S_n\}$, we merge their dependency trees into a single directed graph $\mathcal{G} = (V, E)$. Each unique (lemma, POS) pair is instantiated as a discrete node, ensuring that structural overlap occurs only between syntactically compatible forms:

$$V = \{(l, p) \mid \exists w \in \bigcup_i S_i, \text{lemma}(w) = l, \text{pos}(w) = p\} \quad (1)$$

$$E = \{(u, v, r) \mid \exists \text{ edge } u \xrightarrow{r} v \text{ in any Tree}(S_i)\}$$

where r denotes the dependency relation type. This merged graph supports parameter sharing and structural alignment across sentences, allowing LOGOS to perform cross-sentence syntactic generalization while maintaining the granular distinctions necessary for disambiguating polysemous lemmas.The neuro-symbolic nature of LMDG offers a robust substrate for addressing critical challenges in AI Safety and Alignment. Unlike modern LLMs that rely primarily on probabilistic alignment techniques (e.g., RLHF), which optimize for statistical likelihood rather than hard logical boundaries (Touvron et al., 2023), the LMDG provides an auditable symbolic structure and enables the enforcement of model safety through verifiable structural constraints. We provide more details in Appendix E.

3.1 Quantifying Data Utility and the LMDG Structural Saturation Principle

Unlike standard scaling laws which suggest that model performance scales monotonically with data volume (OpenAI et al., 2023), LMDG allows for a quantitative assessment of the intrinsic utility of new data. We define the **Marginal Information Gain** ($\Delta\mathcal{G}$) as the structural contribution of a new text collection C to the existing LMDG $\mathcal{G}_t$:

$$\Delta\mathcal{G}(C) = \underbrace{|V_C \setminus V_t|}_{\text{New Lemmas}} + \alpha \underbrace{|EC \setminus E_t|}_{\text{New Edges}} \quad (2)$$

where V_C and E_C represent the sets of unique lemmas and edges in the new incoming text. α weighs the relative importance of syntactic diversity (edges) against lexical expansion (nodes). As the LMDG grows, the rate of discovering novel structural triplets (h, d, r) inherently slows. We characterize this progression via **topological velocity** v_{topo}, measuring the rate of graph expansion relative to the number of existing tokens (N):

$$v_{\text{topo}} = \frac{\partial(|V| + \alpha|E|)}{\partial N} \quad (3)$$

We define a **LMDG structural saturation point** as the state where $v_{\text{topo}} < \epsilon$. At this threshold, incoming text contributes negligible new information, signaling that LMDG has captured the representative syntactic distribution of the domain. This provides a principled stopping criterion to avoid computational waste of processing redundant data. As Figure 2 shows, evaluations on BabyLM-100M indicate that LMDG saturates at approximately 300M words assuming no distribution change. More details are provided in Appendix F.

3.2 Topological Compression: Overcoming the Long-Context Bottleneck

LOGOS mitigates the quadratic complexity of attention (Vaswani et al., 2017) by shifting dependency computation from linear sequence length N to the graph's effective diameter. We define operational diameter as the maximum shortest path

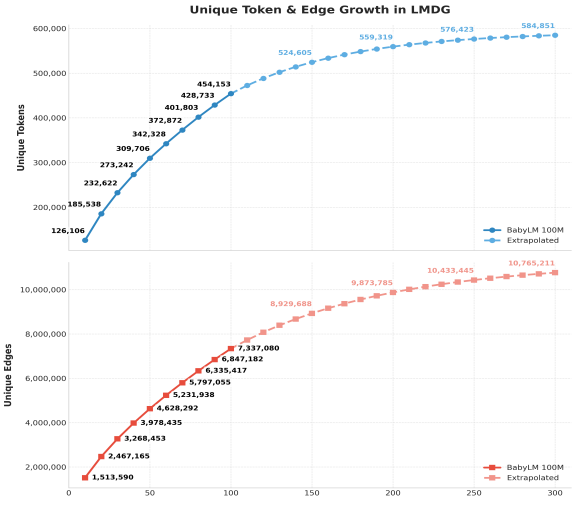

Figure 2: Structural saturation on BabyLM-100M.

between any two nodes in $\mathcal{G}$. LOGOS leverages the small-world properties of linguistic graphs. High-frequency lemmas act as hub nodes, drastically reducing the steps required for global information flow. Mathematically, the graph topology ensures:

$$\text{Diam}(\mathcal{G}) \leq k_{\text{threshold}} \ll N \tag{4}$$

where $k_{\text{threshold}}$ is the maximum hop count before feature homogenization (oversmoothing) occurs. This allows a shallow GCN to achieve global context coverage that would otherwise require $O(N^2)$ attention. This topological compression focuses computation on relational edges, mitigating information decay and lost-in-the-middle phenomenon (Liu et al., 2023).

In summary, LMDG is a neuro-symbolic data structure designed to represent a corpus as a unified global manifold. By shifting from a linear token-based approach to a graph-based topological approach, the LMDG preserves the structural nuances essential for rigorous linguistic research. It serves as a bridge between distributionalist methods and formal structuralism, allowing for mathematical analysis of language through graph-theoretic metrics while maintaining the interpretability of traditional linguistic categories.

4 A GCN-Based Masked Language Model

LOGOS decouples knowledge storage from model parameterization: linguistic structure is preserved in a fixed symbolic memory LMDG discussed in Section 3, while parametric learning is conducted with GCN as shown in Algorithm 2 in Appendix A. Now we will discuss four major components in the LOGOS framework: (1) stochastic multi-mask supervision, (2) graph-based encoding and semantic weighting and fusion, (3) adaptive sampling and contrastive InfoNCE, and (4) constrained/closed-world or unconstrained/open-world prediction.

4.1 Stochastic Multi-Mask Supervision

LOGOS introduces a masking and learning protocol called **Stochastic Multi-Mask Supervision (SMMS)**. This paradigm is specifically designed to account for the structural density of the Lemma-Merged Dependency Graph (LMDG). In our framework, we employ a configurable but uniform masking density where a fixed number of nodes are randomly sampled for occlusion from every sentence.

SMMS ensures that the objective function accounts for consistent relational entropy across the manifold. Mathematically, for a graph with node set V, we define the mask set $\mathcal{M} \subset V$ such that $|\mathcal{M}| = k$. The training objective is to maximize the pseudo-likelihood of the masked nodes given the observed graph topology and edge labels:

$$\mathcal{L}_{\text{SMMS}} = \mathbb{E}_{\mathcal{M} \subset V, |\mathcal{M}|=k} \left[\sum_{v \in \mathcal{M}} -\log P(v \mid \mathcal{G} \setminus \mathcal{M}, \theta) \right] \tag{5}$$

We hypothesize that employing *variable subsets* of nodes—where k is sampled from a dynamic distribution (e.g., $k \sim \text{Poisson}(\lambda)$)—could further enhance the model's ability to generalize across varying levels of syntactic sparsity. This exploration of variable-cardinality masking as a curriculum learning strategy is reserved for future work.

4.2 Graph Encoder and Message Passing

LOGOS encodes LMDG using a K-layer Graph Convolutional Network (GCN) (Kipf and Welling, 2017), which prioritizes structural efficiency and provides understanding how topological connectivity facilitates lexical reconstruction. Let $\mathbf{Z} \in \mathbb{R}^{|V| \times D_{\text{emb}}}$ represent the lemma embeddings initialized as $\mathbf{h}_i^{(0)} = \mathbf{Z}_i$, and these word embeddings can be initialized with a word representation learning method such as Word2Vec (Mikolov et al., 2013).

For each layer $k \in \{0, \ldots, K-1\}$, the node representation is updated via a neighborhood ag-

gregation rule (Gilmer et al., 2017):

$$h_i^{(k+1)} = \sigma\Bigg(W_{\text{self}}^{(k)} h_i^{(k)} + \sum_{j \in \mathcal{N}(i)} \frac{1}{c_{i,j}} W^{(k)} h_j^{(k)} + b^{(k)} \Bigg) \quad (6)$$

where $\mathcal{N}(i)$ denotes the set of immediate neighbors of node i in the LMDG, and $W^{(k)}$ is a learnable weight matrix shared across all edges in layer k. The term $c_{i,j}$ is a normalization constant, typically defined as $\sqrt{d_i d_j}$, where d_i is the degree of node i. This GCN architecture allows LOGOS to capture isotropic structural influence, where the semantic representation of a lemma is refined by the aggregate context of its topological neighbors.

While the current architecture treats all edges as structurally homogeneous, future iterations will implement a Relational GCN by introducing relation-specific weight matrices, so LOGOS can explicitly distinguish between functional roles like *nsubj* and *obj*, providing a granular resolution of predicate-argument structures. Furthermore, the topological properties of the LMDG—such as edge frequency—offer promising avenues for automated noise reduction and domain-specific pruning in downstream applications.

4.3 Semantic Weighting and Fusion

To ground the GCN's contextualized output in a stable lexical space, we apply **Semantic Weighting ($\mathcal{L}_{\text{SEM}}$)**. The final GCN state $h_m^{(K)}$ for a masked node is projected back into the embedding dimension D_{emb} via a projection matrix $\mathbf{W}_{\text{proj}}$:

$$\tilde{\mathbf{z}}'_m = \mathbf{h}_m^{(K)} \mathbf{W}_{\text{proj}} + \mathbf{b}_{\text{proj}} \quad (7)$$

The loss is defined as the Mean Squared Error (MSE) between the predicted vector and the ground-truth lemma embedding $\mathbf{Z}_{y_m}$:

$$\mathcal{L}_{\text{SEM}} = \frac{1}{|\mathcal{M}|} \sum_{m \in \mathcal{M}} \| \tilde{\mathbf{z}}'_m - \mathbf{Z}_{y_m} \|_2^2 \quad (8)$$

To further refine the representation, we utilize **Semantic Fusion** to integrate global context. A tree embedding $\mathbf{h}_{\text{tree}}$ is computed via mean-pooling across all nodes V_s in the sentence: $\mathbf{h}_{\text{tree}} = \frac{1}{|V_s|} \sum_{i \in V_s} \mathbf{h}_i^{(K)}$. The final prediction vector $\mathbf{h}'_m$ is a non-linear fusion of local and global features:

$$\mathbf{h}'_m = \text{ReLU}\left(\begin{bmatrix} \mathbf{h}_m^{(K)} \\ \mathbf{h}_{\text{tree}} \end{bmatrix} \mathbf{W}_{\text{fuse}} + \mathbf{b}_{\text{fuse}} \right) \quad (9)$$

This fusion ensures that LOGOS captures both topological neighborhood of the masked node and broader "thematic gist" of the syntactic tree.

4.4 Adaptive Sampling and InfoNCE

As the LMDG grows, standard softmax over the full vocabulary $|V|$ becomes a computational bottleneck. We alleviate this by implementing adaptive sampling for the output layer (Shi et al., 2025). The symbolic manifold is partitioned into frequency-based clusters, where gradient updates are focused on high-frequency "hub" lemmas while the long-tail lexical periphery is sampled sparsely.

To prevent representation collapse, we supplement the primary objective with a contrastive InfoNCE loss (Oord et al., 2018):

$$\mathcal{L}_{\text{InfoNCE}} = -\mathbb{E}\Bigg[\log \frac{\exp(h'_m \cdot z_{y_m}/\tau)}{\exp(h'_m \cdot z_{y_m}/\tau) + \sum_{j \in \mathcal{N}} \exp(h'_m \cdot z_j/\tau)} \Bigg] \quad (10)$$

Negative samples $\mathcal{N}$ are selected via an adaptive topological strategy that prioritizes lemmas with similar syntactic roles or adjacent neighborhoods in the global LMDG. This contrastive pressure forces the model to learn fine-grained semantic discriminators between nodes that are topologically similar but semantically distinct.

Overall, LOGOS is optimized via a unified multi-task objective that balances structural, semantic, and contrastive supervisions:

$$\mathcal{L}_{\text{Total}} = \mathcal{L}_{\text{CE}} + \lambda_{\text{NCE}} \mathcal{L}_{\text{InfoNCE}} + \lambda_{\text{sem}} \mathcal{L}_{\text{SEM}} \quad (11)$$

4.5 Open-world vs. Closed-world Prediction

During inference LOGOS allows both *Open-world* (unconstrained probabilistic) and *Closed-world* (symbolically constrained) prediction modes. While open-world prediction follows standard maximum likelihood estimation across the entire vocabulary V, closed-world mode leverages the neuro-symbolic property of LMDG to enforce task-specific structural and semantic constraints.

In the closed-world approach for language modeling, we define a composite constraint $C(h, d, r)$ (h: head, d : dependent, r : relation) derived from Universal Dependency guidelines (Nivre et al., 2016). This constraint ensures that any candidate lemma i is linguistically compatible with the masked position m based on its Part-of-Speech, dependency label, and underlying graph topology.

The candidate set $\mathcal{C}_m$ is restricted to include only those lemmas that can functionally replicate the exact neighborhood of the masked node. Specifically, a candidate lemma i is considered valid only if it shares the same set of parent nodes (heads) and children nodes (dependents) as the masked word within the dependency parse tree. Let $\mathcal{P}(m)$ and $\mathcal{D}(m)$ represent the sets of parents and dependents of the masked node, respectively. The set of permissible candidates $\mathcal{C}_m$ is defined as:

$$\mathcal{C}_m = \left\{ i \in V \;\middle|\; \begin{array}{l} \text{Norm}(\tilde{\mathbf{z}}'_m) \cdot \text{Norm}(\mathbf{Z}_i)^\top \geq \xi, \\ C(h, i, r) = 1, \\ \forall p \in \mathcal{P}(m) : (p, i) \in E, \\ \forall d \in \mathcal{D}(m) : (i, d) \in E \end{array} \right\} \tag{12}$$

where $\tilde{\mathbf{z}}'_m$ is the predicted embedding and $\mathbf{Z}_i$ is the embedding of a candidate lemma i. This filtering mechanism ensures that LOGOS does not merely predict a semantically similar word, but one that is structurally interchangeable within the specific syntactic configuration of the sentence.

By pruning the generation space to lemmas that "fit" the LMDG structure, LOGOS reduces the possibility of predicting linguistically incoherent or syntactically impossible candidates—a common failure mode in purely probabilistic models.

For AI safety and alignment, this closed-world constraint mechanism represents a shift from *soft alignment* (probabilistic preference) to *hard alignment* (verifiable rules). Depending on the specific task—ranging from creative generation to formal logic verification—constraint set C can readily incorporate specific requirements. This allows precise fine-grained control to prevent LOGOS from deviating into states that violate predefined logical or linguistic constraints, ensuring that model behavior remains symbolically verifiable.

5 Experiments

Benchmarks and standard splits. We evaluate our model on three standard language modeling benchmarks: Penn Treebank (PTB), WikiText-2 (WT2), and WikiText-103 (WT103) using Pseudo-Perplexity($PPPL$), Top-1 accuracy, and Top-5 accuracy. Details are provided in Appendix D.

Baselines and Evaluation Metrics Our evaluation aims to isolate intrinsic modeling capability of LOGOS using Pseudo-Perplexity (PPPL), a standard intrinsic metric for Masked Language Models (MLMs) (Salazar et al., 2020). A significant challenge in benchmarking MLMs is the scarcity of reported intrinsic metrics; architectures such as RoBERTa or ALBERT are almost exclusively evaluated on downstream transfer tasks (e.g., GLUE), obscuring their fundamental modeling efficiency. To provide a rigorous comparative analysis, we benchmark against well-documented autoregressive baselines where intrinsic Perplexity (PPL) is the established gold standard.

We compare LOGOS against Variational LSTM (Inan et al., 2016), AWD-LSTM (Merity et al., 2018), AWD-LSTM-MoS (Yang et al., 2018), AWD-LSTM-MoS + PDR (Brahma, 2019), Transformer-XL (Dai et al., 2019), Segatron-XL (Bai et al., 2021), and zero-shot GPT-2 (Radford et al., 2019). We acknowledge the theoretical distinction between metrics: autoregressive PPL approximates $P(S)$ via the chain rule $\prod P(w_t|w_{<t})$, whereas PPPL estimates the pseudo-likelihood via $\prod P(w_t|w_{\setminus t})$.[1] By benchmarking against these autoregressive standards, we prioritize a granular, from-scratch evaluation that validates the core LOGOS architecture without the confounding variables of massive-scale pre-training.

LMDG statistics. Table 1 shows statistics of three benchmarks and their LMDGs. To evaluate the sensitivity of LOGOS to parsing quality, we tried two architectural variants of the spaCy pipeline: efficiency-oriented en_core_web_sm (CNN-based) and accuracy-oriented en_core_web_trf (RoBERTa-based). Our empirical results indicate that the performance delta between them is small [2], suggesting that LMDG is robust to minor parsing noise. This justifies the use of high-throughput, lightweight parsers for large-scale LMDG construction, as the global topological properties of the corpus-level manifold effectively mitigate local dependency errors. The parsing time reported in Table 1 corresponds to a single-threaded execution environment, which represents a one-time overhead decoupled from inference phase. Furthermore, since parsing is a highly parallel task and every sentence can be parsed independently, this computational cost is highly scalable. By deploying a distributed MapReduce-style framework across m nodes, the total wall-clock

[1] While mathematically distinct, Wang and Cho (2019) demonstrate that PPPL and PPL are strongly correlated metrics of generative capability ($r > 0.9$ in controlled settings), as both quantify the Kullback-Leibler divergence between the model's distribution and the empirical data distribution.

[2] With WT2, parser en_core_web_sm's results are top1 0.4466, top5 0.6182, ppl 30.82, parser en_core_web_trf's results are top1 0.4472, top5 0.6208, ppl 28.66

Corpus	Number of sentences	Number of total words	Number of nodes	Number of edges	Parsing time (hh:mm:ss)	Graph size(MB)
PTB	44,134	1,170,998	12,083	286,518	em: 00:01:49 trf: 00:16:05	11.80
WT2	93,529	2,611,565	32,449	644,568	em: 00:05:18 trf: 00:56:32	31.69
WT103	3,750,432	100,704,070	322,918	12,369,192	em: 03:17:03 trf: 29:02:06	315.35

Table 1: Details of benchmarks and their LMDGs. Two parsers are sm: en_core_web_sm and trf: en_core_web_trf

Table 2: Results with PTB, WT2, and WT103

Corpus	**Model**	**#Params**	**PPL/ PPPL**
PTB	Variational LSTM	24M	73.2
	AWD-LSTM	24M	57.3
	AWD-LSTM + cache	24M	52.8
	AWD-LSTM + PDR	24.2M	55.6
	AWD-LSTM-MoS	22M	53.8
	GPT-2 small	117M	65.85
	GPT-2 medium	345M	47.33
	GPT-2 large	762M	40.31
	GPT-2 XL	1,542M	35.76
	LOGOS-OpenWorld	10M	20.30
	LOGOS-ClosedWorld	10M	1.48
WT-2	Variational LSTM	24M	87.0
	AWD-LSTM	33M	65.8
	AWD-LSTM + cache	33M	52.0
	AWD-LSTM + PDR	33.6M	63.5
	AWD-LSTM-MoS	35M	60.5
	GPT-2 small	117M	29.41
	GPT-2 medium	345M	22.76
	GPT-2 large	762M	19.93
	GPT-2 XL	1,542M	18.34
	LOGOS-OpenWorld	20M	28.66
	LOGOS-ClosedWorld	20M	1.67
WT-103	GPT-2 small	117M	37.50
	GPT-2 medium	345M	26.37
	GPT-2 large	762M	22.05
	GPT-2 XL	1,542M	17.48
	Transformer XL	151M	24.0
	Transformer XL(large)	257M	18.3
	Segatron-XL (base)	151M	22.5
	Segatron-XL (large)	257M	17.1
	LOGOS-OpenWorld	104M	24.19
	LOGOS-ClosedWorld	104M	1.96

time can be reduced to approximately $1/m$.

Main Results. Table 2 shows that in Logos-OpenWorld setting, the model achieves competitive $PPPL$ scores while utilizing substantially fewer parameters than the autoregressive baselines. This empirical evidence suggests that the graph-based symbolic lexical memory acts as a potent inductive bias, effectively guiding masked word prediction even without candidate space restriction.

In contrast, Logos-ClosedWorld setting yields significantly reduced perplexity values. Consistent with our methodological framing, we interpret these results not as immediate deployment capabilities, but as a diagnostic upper bound. These values isolate the contribution of symbolic memory, demonstrating that when structural constraints are satisfied, the required parametric capacity to recover target tokens is minimal. This diagnostic gap implies that future iterations may progressively relax these constraints by leveraging natural language redundancy to compensate for the absence of syntactic information.

Parameter Efficiency and Scaling Laws As the evaluation scales from PTB to WT-103, the total parameter count increases from 10M to 104M. However, as detailed in Table 3, this growth is dominated by the embedding layer $\mathbf{E} \in \mathbb{R}^{|V| \times d}$, which scales linearly with the vocabulary size $|V|$, rather than the GCN model parameters. This architectural decoupling confirms our central claim: the Graph Convolutional component maintains a relatively static footprint, as symbolic LMDG assumes the burden of storing lexical and relational context.

Detailed results for LOGOS model in Table 3. While PTB and WT-2 permit exhaustive sampling, we observe that restricting masked samples per sentence in WT-103 still yields competitive $PPPL$. This suggests that LOGOS performs efficiently at scale by delegating lexical storage to the explicit graph $\mathcal{G}$, while the parametric GNN (Θ_{GCN}) focuses exclusively on relational inference.

Across all corpora, the LOGOS-CLOSEDWORLD setting achieves high Top-k accuracy. As LMDG size increases, the structural density of $\mathcal{G}$ asymptotically approaches the true linguistic population. In the initial stages of LMDG construction, $\mathcal{G}$ is sparse, with frequent emergence of unseen lemmata and novel dependency relations. As $\mathcal{G}$ accumulates a near-exhaustive inventory of $(\text{lemma}, \text{POS})$ pairs and head-dependent arc types, the representational space becomes more stable, allowing LOGOS to optimize over a saturated symbolic memory.

Corpus	Setting	Samples /sentence	Seconds /epoch	GCN parameters	Embedding parameters	PPPL	Top-1 Acc	Top-5 Acc
PTB	open-world	max	28.8	7,040,418	3,093,248	20.30	0.4879	0.6599
	closed-world	max	31.7	7,044,514	3,093,248	1.48	0.8815	0.9959
WT2	open-world	max	58.1	11,923,130	8,306,944	28.66	0.4472	0.6208
	closed-world	max	68.7	11,927,226	8,306,944	1.67	0.8568	0.9922
WT103	open-world	10	1523.1	21,641,811	82,667,008	24.19	0.4544	0.6504
	closed-world	2	270.1	21,641,811	82,667,008	1.96	0.8288	0.9943

Table 3: Detailed results for LOGOS. "max" means that max/sentence length number of samples are generated for each sentence. The longer epoch time in open-world WT-103 is due to more samples generated from each sentence.

Impact of Masking Rate. Table 4 illustrates the performance sensitivity to the masking rate $\mu \in \{1, 2, 3\}$ words per sentence. Given an average sentence length $\bar{L} \approx 20$, a rate of $\mu = 3$ approximates the industry standard of 15% typically utilized in MLM pre-training (Devlin et al., 2019). While an increased μ intuitively elevates the difficulty of the reconstruction task by reducing local context, the difficulty can be partially compensated by generating more samples from each sentence. Our results indicate that the benefit of higher sample throughput outweighs the increased predictive entropy. Additional relational constraints activated in LMDG during multi-word masking provide a richer gradient for GCN encoder. LOGOS is robust to standard masking densities and benefits from the combinatorial diversity of larger masking sets.

Masking rate	Samples/ sentence	PPPL	Top-1 accu.	Top-5 accu.
2	100	35.84	42.06	60.31
2	180	34.27	42.54	61.05
3	100	42.45	39.55	58.53
3	160	37.67	40.08	59.57

Table 4: Impact of masking rate using WikiText-2

Ablation study. Table 5 validates LOGOS' components. Removing Adaptive Sampling causes the sharpest performance drop (PPPL +2.65, Top-1 -3.37%). Omitting Semantic Constraints or InfoNCE Loss also degrades results, confirming that both are essential for optimal performance. Full configuration consistently achieves the best metrics.

Model robustness and training stability. As Figure 3 shows, LOGOS training is quite stable and robust, and usually converges at around 60 epochs.

Setting	PPPL	Top1 acc.	Top5 acc.
Original	20.30	48.79	65.99
No InfoNCE	21.92	48.20	65.84
No Semantic	21.66	47.49	65.51
No Ad. Sam.	22.95	45.42	64.74

Table 5: Ablation study using PTB

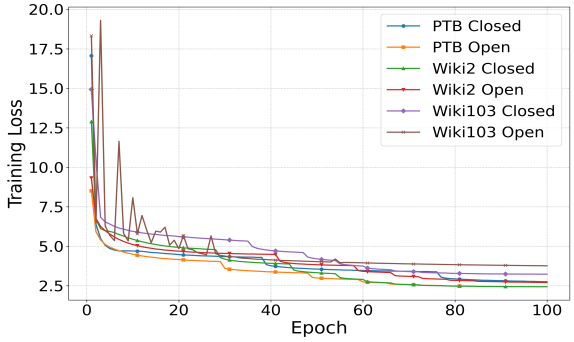

Figure 3: Training stability and convergence

6 Conclusion

This work introduced LOGOS, a neuro-symbolic framework that re-architects language modeling by replacing global quadratic attention with a Lemma-Merged Dependency Graph (LMDG) and a GNN-based encoder. By grounding neural representations in a shared symbolic manifold, LOGOS achieves structural generalization with high parameter efficiency, effectively decoupling lexical storage from parametric inference. Our results demonstrate that topological compression—leveraging the small-world properties of linguistic graphs—successfully mitigates the long-context bottleneck without the instability of deep GCNs or the overhead of traditional scaling laws. Beyond architectural efficiency, symbolic nature of LOGOS offers a path toward verifiable AI safety, which we will explore in future work.

7 Limitations

Despite the performance, efficiency, and interpretability gains offered by LOGOS, several limitations must be acknowledged.

7.1 Dependency Parsing Overhead and Noise

The primary structural requirement of our framework is the use of an external dependency parser to construct the LMDG. Current state-of-the-art parsers are not 100% accurate; consequently, parsing errors—such as misidentified heads or incorrect relation labels—can introduce noise into the symbolic manifold.

7.2 Vocabulary Scalability and Interpretability Trade-offs

Unlike traditional Transformers that utilize subword tokenization (e.g., BPE) to manage vocabulary size, LOGOS operates on a per-lemma basis. This results in a significantly larger vocabulary; however, this is a deliberate design choice intended to preserve the one-to-one mapping between nodes and symbolic lemmas, ensuring maximum interpretability. While a large vocabulary typically presents a scaling challenge, our experiments on WikiText-103 demonstrate that the architecture remains robust even with a vocabulary of more than 260,000 unique lemmas. This scalability is facilitated by the effective adaptive sampling strategy. Given that the number of unique lemmas in natural language is naturally bounded, the "large vocabulary" problem remains functionally manageable.

7.3 Syntactic Sparsity

Finally, the model's reliance on syntactic structure may result in performance degradation when processing highly informal or ungrammatical text where a dependency tree cannot be reliably formed.

References

Anonymous. 2025. Grounding generative planners in verifiable logic: A hybrid architecture for trustworthy embodied AI. In *Submitted to The Fourteenth International Conference on Learning Representations*. Under review.

He Bai, Peng Shi, Jimmy Lin, Yuqing Xie, Luchen Tan, Kun Xiong, Wen Gao, and Ming Li. 2021. Segatron: Segment-aware transformer for language modeling and understanding. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 35, pages 12526–12534.

Ali Behrouz, Meisam Razaviyayn, Peilin Zhong, and Vahab Mirrokni. 2025. Nested learning: The illusion of deep learning architectures. In *The Thirty-ninth Annual Conference on Neural Information Processing Systems*.

Siddhartha Brahma. 2019. Improved language modeling by decoding the past. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 1468–1476, Florence, Italy. Association for Computational Linguistics.

Ziwei Chai, Tianjie Zhang, Liang Wu, Kaiqiang Han, Xiaohai Hu, Xuanwen Huang, and Yang Yang. 2025. Graphllm: Boosting graph reasoning ability of large language model. *IEEE Transactions on Big Data*, pages 1–9.

Zihang Dai, Zhilin Yang, Yiming Yang, Jaime Carbonell, Quoc Le, and Ruslan Salakhutdinov. 2019. Transformer-XL: Attentive language models beyond a fixed-length context. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 2978–2988, Florence, Italy. Association for Computational Linguistics.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of NAACL-HLT*, pages 4171–4186.

Daniel Gildea and Daniel Jurafsky. 2002. Automatic labeling of semantic roles. *Computational Linguistics*, 28(3):245–288.

Justin Gilmer, Samuel S Schoenholz, Patrick F Riley, Oriol Vinyals, and George E Dahl. 2017. Neural message passing for quantum chemistry. In *ICML*.

Albert Gu and Tri Dao. 2024. Mamba: Linear-time sequence modeling with selective state spaces. *First Conference on Language Modeling*.

Hakan Inan, Khashayar Khosravi, and Richard Socher. 2016. Tying word vectors and word classifiers: A loss framework for language modeling. *CoRR*, abs/1611.01462.

Ziwei Ji, Nayeon Lee, Rita Frieske, Tiezheng Yu, Dan Su, Yan Xu, Etsuko Ishii, Ye Jin Bang, Andrea Madotto, and Pascale Fung. 2023. Survey of hallucination in natural language generation. *ACM Comput. Surv.*, 55(12).

Thomas N. Kipf and Max Welling. 2017. Semi-supervised classification with graph convolutional networks. In *International Conference on Learning Representations (ICLR)*.

Omer Levy and Yoav Goldberg. 2014. Dependency-based word embeddings. In *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 302–308.

Shucheng Li, Lingfei Wu, Shiwei Feng, Fangli Xu, Fengyuan Xu, and Sheng Zhong. 2020. Graph-to-tree neural networks for learning structured input-output translation with applications to semantic parsing and math word problem. In *Findings of the Association for Computational Linguistics: EMNLP 2020*, pages 2841–2852, Online. Association for Computational Linguistics.

Tianyi Li, Mingda Chen, Bowei Guo, and Zhiqiang Shen. 2025. A survey on diffusion language models. *Preprint*, arXiv:2508.10875.

Nelson F. Liu, Kevin Lin, John Hewitt, Ashwin Paranjape, Michele Bevilacqua, Fabio Petroni, and Percy Liang. 2023. Lost in the middle: How language models use long contexts. *Preprint*, arXiv:2307.03172.

Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. 2019. Roberta: A robustly optimized bert pretraining approach. In *arXiv preprint arXiv:1907.11692*.

Diego Marcheggiani and Ivan Titov. 2017. Encoding sentences with graph convolutional networks for semantic role labeling. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 1506–1515.

Mitchell P. Marcus, Beatrice Santorini, and Mary Ann Marcinkiewicz. 1993. Building a large annotated corpus of English: The Penn Treebank. *Computational Linguistics*, 19(2):313–330.

Stephen Merity, Nitish Shirish Keskar, and Richard Socher. 2018. Regularizing and optimizing lstm language models. In *Proceedings of ICLR*.

Stephen Merity, Caiming Xiong, James Bradbury, and Richard Socher. 2016. Pointer sentinel mixture models. *arXiv preprint*. ArXiv:1609.07843.

Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. 2013. Efficient estimation of word representations in vector space. *Preprint*, arXiv:1301.3781.

Tomáš Mikolov, Martin Karafiát, Lukáš Burget, Jan Černockỳ, and Sanjeev Khudanpur. 2010. Recurrent neural network based language model. In *Interspeech*.

Joakim Nivre, Marie-Catherine de Marneffe, Filip Ginter, Yoav Goldberg, Jan Hajič, Christopher D. Manning, Ryan McDonald, Slav Petrov, Sampo Pyysalo, Natalia Silveira, Reut Tsarfaty, and Daniel Zeman. 2016. Universal Dependencies v1: A multilingual treebank collection. In *Proceedings of the Tenth International Conference on Language Resources and Evaluation (LREC'16)*, pages 1659–1666, Portorož, Slovenia. European Language Resources Association (ELRA).

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

OpenAI, Josh Achiam, and et al. 2023. Gpt-4 technical report. *arXiv preprint arXiv:2303.08774*.

Bo Peng, Ruichong Zhang, Daniel Goldstein, Eric Alcaide, Xingjian Du, Haowen Hou, Jiaju Lin, Jiaxing Liu, Janna Lu, William Merrill, Guangyu Song, Kaifeng Tan, Saiteja Utpala, Nathan Wilce, Johan S. Wind, Tianyi Wu, Daniel Wuttke, and Christian Zhou-Zheng. 2025. RWKV-7 "goose" with expressive dynamic state evolution. In *Second Conference on Language Modeling*.

Matthew E. Peters, Mark Neumann, Robert Logan, Roy Schwartz, Vidur Joshi, Sameer Singh, and Noah A. Smith. 2019. Knowledge enhanced contextual word representations. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 43–54, Hong Kong, China. Association for Computational Linguistics.

Yujie Qian, Enrico Santus, Zhijing Jin, Jiang Guo, and Regina Barzilay. 2019. GraphIE: A graph-based framework for information extraction. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 751–761, Minneapolis, Minnesota. Association for Computational Linguistics.

Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, and Ilya Sutskever. 2019. Language models are unsupervised multitask learners. Technical report, OpenAI. Technical report.

Julian Salazar, Davis Liang, Toan Q. Nguyen, and Katrin Kirchhoff. 2020. Masked language model scoring. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 2699–2712, Online. Association for Computational Linguistics.

Abulhair Saparov, Richard Yuanzhe Pang, Vishakh Padmakumar, Nitish Joshi, Seyed Mehran Kazemi, Najoung Kim, and He He. 2023. Testing the general deductive reasoning capacity of large language models using ood examples. In *Proceedings of the 37th International Conference on Neural Information Processing Systems*, NIPS '23, Red Hook, NY, USA. Curran Associates Inc.

Yikang Shen, Shawn Tan, Alessandro Sordoni, and Aaron Courville. 2019. Ordered neurons: Integrating tree structures into recurrent neural networks. In *Proceedings of the International Conference on Learning Representations (ICLR)*.

Zhenning Shi, Yijia Zhu, Yi Xie, Junhan Shi, Guorui Xie, Haotian Zhang, Yong Jiang, Congcong Miao, and Qing Li. 2025. Reasoning under uncertainty: Efficient LLM inference via unsupervised confidence

dilution and convergent adaptive sampling. In *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing*, pages 32192–32206, Suzhou, China. Association for Computational Linguistics.

Richard Socher, Alex Perelygin, Jean Wu, Jason Chuang, Christopher D. Manning, Andrew 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, Seattle, Washington, USA. Association for Computational Linguistics.

Linfeng Song, Yue Zhang, Zhiguo Wang, and Daniel Gildea. 2018. A graph-to-sequence model for AMR-to-text generation. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1616–1626, Melbourne, Australia. Association for Computational Linguistics.

Kai Sheng Tai, Richard Socher, and Christopher D. Manning. 2015. Improved semantic representations from tree-structured long short-term memory networks. In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics*, pages 1556–1566.

Hugo Touvron, Louis Martin, Kevin Stone, Peter Albert, Amjad Almahairi, Yasmine Babaei, Nikolay Bashlykov, Soumya Batra, Prajjwal Bhargava, Shruti Bhosale, Dan Bikel, Lukas Blecher, Cristian Canton-Ferrer, Moya Chen, Guillem Cucurull, David Esiobu, Jude Fernandes, Jeremy Fu, Wenyin Fu, and 49 others. 2023. Llama 2: Open foundation and fine-tuned chat models. *CoRR*, abs/2307.09288.

Shikhar Vashishth, Manik Bhandari, Prateek Yadav, Piyush Rai, Chiranjib Bhattacharyya, and Partha Talukdar. 2019. Incorporating syntactic and semantic information in word embeddings using graph convolutional networks. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 3308–3318, Florence, Italy. Association for Computational Linguistics.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *Proceedings of the 31st International Conference on Neural Information Processing Systems*, NIPS'17, page 6000–6010, Red Hook, NY, USA. Curran Associates Inc.

Petar Veličković, Guillem Cucurull, Arantxa Casanova, Adriana Romero, Pietro Liò, and Yoshua Bengio. 2018. Graph attention networks. In *International Conference on Learning Representations (ICLR)*.

Alex Wang and Kyunghyun Cho. 2019. BERT has a mouth, and it must speak: BERT as a Markov random field language model. In *Proceedings of the Workshop on Methods for Optimizing and Evaluating Neural Language Generation*, pages 30–36, Minneapolis, Minnesota. Association for Computational Linguistics.

Lingfei Wu, Yu Chen, Kai Shen, Xiaojie Guo, Hanning Gao, Shucheng Li, Jian Pei, and Bo Long. 2021. Graph neural networks for natural language processing: A survey. *CoRR*, abs/2106.06090.

Zhilin Yang, Zihang Dai, Ruslan Salakhutdinov, and William W. Cohen. 2018. Breaking the softmax bottleneck: A high-rank RNN language model. In *International Conference on Learning Representations*.

Jiawei Zhang, Haopeng Zhang, Congying Xia, and Li Sun. 2020. In *Graph-Bert: Only Attention is Needed for Learning Graph Representations*, volume abs/2001.05140. [link].

Zhuosheng Zhang, Yuwei Wu, Junru Zhou, Sufeng Duan, and Hai Zhao. 2019. Sg-net: Syntax-guided machine reading comprehension. volume abs/1908.05147.

Jinman Zhao, Gerald Penn, and Huan Ling. 2021. Structural realization with GGNNs. In *Proceedings of the Fifteenth Workshop on Graph-Based Methods for Natural Language Processing (TextGraphs-15)*, pages 115–124, Mexico City, Mexico. Association for Computational Linguistics.

Appendix

A LOGOS Algorithm

A.1 LMDG Construction: From Linear Sequences to Symbolic Manifolds

Algorithm 1 LOGOS Phase I: Construct Lemma-Merged Dependency Graph (LMDG)

Input: Parsed corpus $\mathcal{D}$
Output: LMDG Θ
Initialize symbolic graph $\mathcal{G} = (V, E)$
for each sentence $s \in \mathcal{D}_{\text{train}}$ **do**
  for each token (ℓ, u) in s **do**
    if $u \neq$ PUNCT **then**
      Add or retrieve lemma node $\ell\_u \in V$
    end if
  end for
  for each dependency $(h \to d, r)$ in s **do**
    Add labeled edge (h, d, r) to E
  end for
end for

The construction of the **Lemma-Merged Dependency Graph (LMDG)** represents the foundational architectural shift in LOGOS, moving from

linear, sequence-based processing to a global, topological representation. This phase transforms the training corpus into a unified symbolic manifold where lexical units are anchored by their functional syntactic roles.

A.1.1 Structural Logic and Procedural Walkthrough

As detailed in Algorithm 1, the construction process distills a parsed corpus into two distinct stages of symbolic unification:

1. **Lemma-POS Node Fusion (Lines 4–8):** Unlike subword tokenization schemes (e.g., BPE) which fragment semantic meaning, LOGOS utilizes **lemmatization** combined with **Part-of-Speech (POS)** anchoring.
    - *Morphological Collapse:* Variations such as *eating, ate,* and *eats* are mapped to a single semantic root (*eat*).
    - *Functional Disambiguation:* By appending the POS tag u to the lemma ℓ, the algorithm creates distinct nodes for homonyms with different syntactic profiles (e.g., bank_NOUN vs. bank_VERB).
2. **Global Dependency Merging (Lines 10–12):** The core structural innovation lies in the global merging of local sentence trees. If disparate sentences share a syntactic relationship $(h \xrightarrow{r} d)$, they update a single, persistent edge in the global manifold. This turns the LMDG into a multi-relational graph where edges are explicitly typed by their **Universal Dependency (UD)** relations.

A.1.2 Discussion: Topological Advantages for Language Modeling

The LMDG serves as a form of **topological compression**. While traditional Transformers must rederive the relationship between words at every occurrence via self-attention, the LMDG explicitly stores these relationships within a shared geometry.

- **Small-World Connectivity:** The merging of common lemmas creates high-degree "hub nodes" (e.g., auxiliary verbs and common nouns). This significantly reduces the average path length across the lexical manifold, allowing a Graph Convolutional Network (GCN) to aggregate global contextual information through a limited number of message-passing steps.
- **Syntactic Inductive Bias:** By restricting message passing to explicit dependency edges, the LMDG acts as a **hard attention mechanism**. This filters out the spurious correlations often found in the $O(N^2)$ quadratic attention of Transformers, focusing the model's energy on linguistically grounded relational dependencies.
- **Linguistic Interpretability:** Because every node and edge corresponds to a formal category (Lemma and UD Relation), the internal representations of the model are directly mapable to human grammar, providing a transparent substrate for analyzing how the model learns syntax.

A.2 Phase II: Representation Learning and Masked Words Prediction

While Phase I establishes the LMDG manifold, Phase II (Algorithm 2) defines the neural learning process. This stage utilizes a Graph Convolutional Network (GCN) to propagate information across the global topology, supervised by a novel multi-masking protocol.

A.2.1 Stochastic Multi-Mask Supervision (SMMS)

Unlike standard Masked Language Modeling (MLM) which masks single tokens in a linear string, the **SMMS** protocol (Lines 7–12) operates on the sentence-level dependency subgraph $\mathcal{G}_s$.

- *Structural Erasure:* By sampling a mask set $\mathcal{M}_s$ of size k and removing these nodes from the graph, the model is forced to reconstruct the missing lemma using only the remaining topological context.
- *Multi-Hop Dependency:* Since nodes in the LMDG are connected by syntactic relations rather than proximity, the GCN must learn to aggregate information across multiple "hops" to resolve the identity of a masked node, effectively learning the underlying logic of the Universal Dependency rules.

A.2.2 The Hybrid Loss Objective

To ensure the learned embeddings are both syntactically accurate and semantically rich, LOGOS employs a tripartite loss function (Lines 18–21):

Algorithm 2 LOGOS Phase II: Train GCN over LMDG

Initialize GCN parameters Θ
Initialize embeddings $\mathbf{Z}$ with Word2Vec
for each training epoch **do**
  for each minibatch $\mathcal{B} \subset \mathcal{D}$ **do**
    Initialize masked example set $\mathcal{M} \leftarrow \emptyset$
    for each sentence graph $\mathcal{G}_s \in \mathcal{B}$ **do**
      Sample mask set $\mathcal{M}_s \subset V_s$ with $|\mathcal{M}_s| = k$
      for each masked node $m \in \mathcal{M}_s$ **do**
        Remove m from $\mathcal{G}_s$
        Add masked graph to $\mathcal{M}$
      end for
    end for
    Encode $\mathcal{M}$ using K-layer GCN over $\mathcal{G}$
    Obtain contextual node representations $\mathbf{h}^{(K)}$
    Compute local masked-node embeddings and global tree embeddings
    Fuse symbolic context with parametric representations
    Compute cross-entropy loss $\mathcal{L}_{\text{CE}}$
    Compute semantic alignment loss $\mathcal{L}_{\text{SEM}}$
    Compute contrastive InfoNCE loss $\mathcal{L}_{\text{NCE}}$
    $\mathcal{L} \leftarrow \mathcal{L}_{\text{CE}} + \lambda_{\text{sem}}\mathcal{L}_{\text{SEM}} + \lambda_{\text{NCE}}\mathcal{L}_{\text{NCE}}$
    Update Θ via backpropagation
  end for
end for
/* Inference: open-world vs. closed-world decoding */
Prediction with either (i) open-world unconstrained softmax over V, or (ii) closed-world decoding constrained by symbolic rules

1. **Cross-Entropy Loss ($\mathcal{L}_{\text{CE}}$):** The primary objective for lemma prediction, measuring the model's ability to recover the exact identity of the masked node m from the vocabulary V.

2. **Semantic Alignment Loss ($\mathcal{L}_{\text{SEM}}$):** This term regularizes the contextual representations $\mathbf{h}^{(K)}$ against the Word2Vec-initialized embeddings $\mathbf{Z}$. It ensures that the GCN's dynamic outputs do not drift too far from the established semantic manifold.

3. **Contrastive InfoNCE Loss ($\mathcal{L}_{\text{NCE}}$):** To sharpen the discriminative power of the model, we use Noise-Contrastive Estimation. This forces the model to maximize the similarity between the masked representation and the ground-truth lemma, while minimizing similarity with negative samples (distractor lemmas), effectively pushing unrelated concepts apart in the embedding space.

A.2.3 Inference: Neuro-Symbolic Decoding

The algorithm concludes with a flexible inference strategy. In **Open-world** mode, the model functions as a traditional probabilistic predictor. However, in **Closed-world** mode, the symbolic constraints $C(h, d, r)$ (h: head, d : dependent, r : relation) derived from the LMDG act as a decoder filter. This ensures that the predicted lemma is not only semantically plausible but also structurally valid within the target dependency tree, effectively reducing ungrammatical hallucinations.

B Hardware and software environments

Component	Specification
CPU	2× Intel Xeon Gold 6326 @ 2.90GHz
GPU	4× NVIDIA L40S
OS	Linux (x86_64)
Python	3.10.19
Framework	PyTorch 2.4.1 (CUDA 12.4)
Parser	PTB: spaCy `en_core_web_sm` WikiText-2/103: spaCy `en_core_web_trf`

Table 6: Hardware, software, and parser environments.

C Training and hyperparameter setting

We evaluate our model on the Penn Treebank (PTB), WikiText-2, and WikiText-103 datasets under both Closed-world and Open-world settings.

Across all experiments, we maintain a consistent Graph Convolutional Network (GCN) architecture with 4 layers and a hidden dimension of 512. Optimization is performed using a learning rate of 1×10^{-3}, a dropout rate of 0.1. We use 4 NVIDIA L40S GPUs to run each model. We utilize an adaptive softmax mechanism to handle large vocabularies efficiently. The comprehensive dataset-specific hyperparameters and model statistics are detailed in Table 7.

Each node is represented by a learnable embedding. A 4-layer GCN propagates information across the lexical graph. Dropout of 0.1 is applied to node embeddings to prevent overfitting. We set the patience equal to the maximum number of epochs (100) for PTB and WikiText-2 to ensure

Table 7: Detailed hyperparameter settings for PTB, WikiText-2, and WikiText-103 under Closed and Open world settings. Common parameters across single masked models include: GCN Hidden Dim = 512, Num GCN Layers = 4, Dropout = 0.1, Learning Rate = 0.001, Eval Temperature = 1.0, Eval Top-k = 0, Eval Top-p = 0.0.

Parameter	PTB		WikiText-2		WikiText-103	
	Closed	Open	Closed	Open	Closed	Open
Train Masks / Sent (N)	1000	1000	1000	1000	2	10
Eval Masks / Sent	1	1	1	1	1	1
Train Batch Size	1024	1024	1024	1024	4096	5000
Val Batch Size	64	64	64	64	32	64
Test Batch Size	64	64	64	64	32	64
Weight Decay	8×10^{-5}	8×10^{-5}	8×10^{-5}	1×10^{-4}	8×10^{-5}	1×10^{-5}
Max Epochs	100	100	100	100	100	500
Adapt. SM Cutoffs	[8k, 9k]	[8k, 9k]	[15k, 25k]	[15k, 25k]	[20k, 40k, 200k]	[20k, 40k, 200k]
Total Params	10,137,762	10,133,666	20,234,170	20,230,074	104,308,819	104,308,819
Graph Mem (MB)	11.80	11.80	31.69	31.69	315.35	315.35

the full training trajectory. For WikiText-103, we increase the training epochs to 500 to handle the large-scale dataset.

D Benchmarks, standard splits, and evaluation metrics

Benchmarks and standard splits. We evaluate our model on three standard language modeling benchmarks: Penn Treebank (PTB), WikiText-2 (WT2), and WikiText-103 (WT103). The PTB dataset (Marcus et al., 1993), in its widely used pre-processed form (Mikolov et al., 2010), consists of approximately 1.0M tokens (929k train, 73k validation, 82k test) with a vocabulary limited to the 10k most frequent words. WikiText-2 and WikiText-103 (Merity et al., 2016) are sourced from verified "Good" and "Featured" Wikipedia articles, preserving casing, punctuation, and numbers. WT2 is a mid-sized corpus containing 2.1M train, 217k validation, and 245k test tokens with a vocabulary of 33k. WT103 is a significantly larger long-term dependency benchmark comprising 103M training tokens across 28k articles, with a vocabulary of 267k and standard splits of 217k validation and 245k test tokens. We preprocess these corpora using the **spaCy** dependency parser (Table 6), utilizing the en_core_web_sm model for PTB and the transformer-based en_core_web_trf model for WikiText-2 and WikiText-103 to ensure high-fidelity structural graphs.

Evaluation Metrics. We evaluate LOGOS with the following metrics:

- **Pseudo-Perplexity(PPPL):** Standard language modeling metric over unmasked tokens:

$$PPPL(W) = \sum_{i=1}^{|W|} \log P(w_i | W_{\setminus i}) \quad (13)$$

- **Top-k Masked Accuracy:** Measures whether the gold token appears among the top-k predictions:

$$\text{Top-}k \text{ Accuracy} = \frac{1}{N} \sum_{i=1}^{N} \mathbf{1}\{w_i \in \text{top-}k(\hat{w}_i)\} \quad (14)$$

  We use $k = 1$ and $k = 5$.

Prediction is made and pseudo-perplexity is computed on lemma-typed forms, not on surface tokens.

E Potential for Verifiable AI Safety, Alignment, Hallucination Mitigation

Modern LLMs rely primarily on probabilistic alignment techniques (e.g., RLHF), which optimize for statistical likelihood rather than hard logical boundaries (Touvron et al., 2023). While this paper focuses on language modeling, the LMDG architecture inherently provides a symbolic substrate for alignment and hallucination research. By manipulating the graph topology, LMDG can provide some new options for enforcing model safety. Specifically in LMDG message passing is explicitly governed by the edge set E. This provides a mechanism for *Semantic Isolation*: hypothetical pruning of subgraphs $\mathcal{G}_{\text{unsafe}} \subset \mathcal{G}$ representing hazardous or non-aligned conceptual pathways. By enforcing

a zero-weight constraint on edges between sensitive lemma clusters, one could theoretically ensure that a hidden representation h_i cannot aggregate information from prohibited regions of the manifold. Although these symbolic "firewalls" offer a promising path toward verifiable AI, they remain architectural concepts. Further research is required to evaluate the trade-off between strict constraint adherence and the fluidity of linguistic generation.

To address compositional hallucinations (Ji et al., 2023)—where the model incorrectly assembles valid but unrelated entities—we propose anchoring predictions to the global edge set E_{global}. Rather than validating components in isolation, we define a **Support Function** $S(h, r, t)$ that treats the triplet (h, r, t) as an indivisible atomic unit:

$$S(h, r, t) = \begin{cases} 1 & \text{if } (h, r, t) \in E_{\text{global}} \\ 0 & \text{otherwise} \end{cases} \tag{15}$$

During inference, a factuality penalty $\mathcal{L}_{\text{fact}}$ is applied to suppress "stochastic stitching," ensuring the model only generates relations witnessed within the symbolic manifold:

$$\mathcal{L}_{\text{fact}} = \gamma \cdot \max(0, \sigma - S(h, r, t)) \tag{16}$$

where σ is a threshold and γ is a scaling factor. This mechanism ensures internal consistency by requiring that the specific head-tail connection via relation r exists in the global structural consensus, preventing the formation of spurious edges from high-probability but disconnected nodes.

F Study of LMDG structural saturation with BabyLM-100M

In the analysis of Language Modeling via Lemma-Merged Dependency Graphs (LMDG) structural saturation using the BabyLM corpus, the methodology rests upon two foundational assumptions:

1. **Information Incrementality via Discrete Syntactic Units:** Currently LMDG quantifies "novel information" strictly through the lens of unseen lemmata and novel dependency arcs (head-dependent pairs). This excludes surface-level morphological variations, repetitive n-gram patterns, lemma frequency, or dependency arc frequency information, focusing instead on the expansion of the lexico-semantic frontier and the syntactic manifold.

2. **Stationarity in Textual Distribution:** When extrapolating structural growth from actual BabyLM to larger corpus, it is assumed that the underlying generative distribution $\mathcal{P}$ (the "textual manifold") remains invariant. This implies that the statistical properties of syntax—such as the power-law distribution of dependency and vocabulary growth rates—persist as the sample size N increases:

$$\mathcal{P}(\text{syntax} \mid N_{\text{BabyLM}}) \approx \mathcal{P}(\text{syntax} \mid N_{\infty})$$

The BabyLM Challenge (https:babylm.github.io) is a benchmark focused on sample-efficient language acquisition, so it fits our goal of analyzing LMDG structural saturation point. While mainstream Large Language Models (LLMs) are trained on trillions of tokens, BabyLM restricts models to a "human-scale" dataset of approximately 10^8 words—comparable to the linguistic input a child receives by age 13. The corpus includes transcribed speech, children's stories, and educational materials to simulate naturalistic human input rather than generic web scrapes. LMDG maps text to a dependency graph. In this context, structural saturation in LMDG refers to the convergence point where LMDG has mapped the exhaustive set of lemmata and dependency relations possible within a natural language.

Here is the process of our LMDG structural saturation study. The BabyLM-100M corpus is partitioned into ten disjoint subsets, denoted as chunks $\{c_1, c_2, \ldots, c_{10}\}$, where each chunk $|c_i| \approx 10^7$ words. To analyze the growth of syntactic structures, we employ an accumulated sampling approach. Let D_k be the accumulated corpus at step k, such that:

$$D_k = \bigcup_{i=1}^{k} c_i, \quad \text{for } k \in \{1, \ldots, 10\}$$

For each D_k, we compute the set of unique lemmata $\mathcal{L}(D_k)$ and the set of unique dependency relations $\mathcal{R}(D_k)$. The total count of unique structural units U_k is defined as the cardinality of their union:

$$U_k = |\mathcal{L}(D_k) \cup \mathcal{R}(D_k)|$$

Empirical observation of the sequence $\{U_1, \ldots, U_{10}\}$ indicates that the structural saturation point (where $U_k \approx U_{k-1}$) is not reached

within the BabyLM-100M. Consequently, we apply the *Distributional Stationarity* assumption to model the growth curve.

We extrapolate the trajectory of U using a non-linear growth function to estimate the structural diversity up to a corpus size of $N = 300$M words:

$$\hat{U}(N) = \gamma N^{\beta}, \quad \text{for } 100\text{M} < N \leq 300\text{M}$$

where γ and β are parameters estimated from the observed BabyLM-100M distribution, where $\gamma = 1.23$ and $\beta = 0.56$. The result is shown in Figure 2. At the corpus size of 300M words, based on the definition of v_{topo} in Equation (4), assume $\alpha = 1.0$ and $\epsilon = 0.025$. Since $v_{\text{topo}} = 0.023$ and is less than ϵ, LMDG reaches structural saturation point at the corpus size of $N = 300$M words.

G AI Assistants In Research Or Writing

We used LLMs to help with coding, literature review, and polishing writing.