

From Attention to Gluing: A Sheaf-State Architecture for Lower-Complexity Language Models

Juan J. Segura

`juan.segura.f@unab.cl`

Universidad Andrés Bello

Research Article

Keywords: language models, Transformer attention, sheaf theory, categorical deep learning, state-space models, sparse computation, mathematical artificial intelligence.

Posted Date: September 2nd, 2026

DOI: <https://doi.org/10.21203/rs.3.rs-10204001/v1>

License: © ⓘ This work is licensed under a Creative Commons Attribution 4.0 International License.

[Read Full License](#)

Additional Declarations: No competing interests reported.

From Attention to Gluing: A Sheaf-State Architecture for Lower-Complexity Language Models

Juan Segura

Facultad de Ingenieria, Universidad Andres Bello, Chile

juan.segura.f@unab.cl

ORCID:0009-0004-2742-435

Abstract

Dense self-attention is the architectural substrate behind modern Transformer language models. Its strength is that it avoids premature structural commitments: every token can interact with every other token, allowing syntax, reference, instruction relevance, discourse continuity, and task dependence to be learned over a common relational substrate. The same feature is also computationally excessive. Dense self-attention treats the context as a complete token site even when the useful linguistic and inferential relations are sparse, typed, hierarchical, and state-dependent. This paper develops a categorical diagnosis of this overcompleteness and proposes a Sheaf-State Language Model: a lower-complexity architecture in which local state-space dynamics maintain typed context patches and sparse gluing morphisms exchange information only across learned compatible overlaps. We do not claim a trained foundation model. Instead, we provide a formal architecture, complexity propositions, and synthetic dependency simulations showing that complete attention achieves maximal dependency coverage at asymptotically maximal relational waste, while sparse typed gluing can recover designed long-range dependencies with far fewer active morphisms. The contribution is a theory-and-simulation program for replacing the complete attention site by a learned context topology.

Keywords: language models; Transformer attention; sheaf theory; categorical deep learning; state-space models; sparse computation; mathematical artificial intelligence.

1 Introduction

The Transformer architecture made attention the central sequence operation by dispensing with recurrence and convolution in favor of multi-head self-attention [2]. Its success can be understood as a deliberate form of relational overprovision: the model does not need to know in advance whether a token relation is syntactic, semantic, referential, instructional, or inferential. It can offer all token-token relations as candidates and learn compatibility weights from data.

This paper argues that this overcomplete relational substrate is both the reason for the Transformer’s flexibility and a major source of its hardware cost. Dense self-attention gives high coverage of possible context relations, but the underlying complete graph is usually much richer than the sparse dependency structure required by a particular linguistic or reasoning operation. We therefore ask whether the primitive relational object should be changed. Rather than approximating full attention, we propose replacing the complete token site by a learned sparse site of typed context patches.

The proposed model is not an empirical replacement for current large language models. It is a formal architectural hypothesis. It combines four existing directions—distributed neural language

modeling [1], state-space sequence modeling [7], sparse expert activation [6], and low-bit morphisms [10]—under a categorical principle: meanings should be maintained as compatible local sections over a structured context site, not repeatedly reconstructed through a complete graph of token-token interactions.

2 Related Work and Differentiation

This section clarifies that the present proposal is not isolated. Several nearby literatures already contain important parts of the architecture: neural sheaf models, attention over topological structures, efficient state-space sequence models, sparse experts, and low-bit language models. The contribution here is not to introduce these components independently, but to use them to formulate a specific diagnosis of dense self-attention and a corresponding language-model architecture.

2.1 Neural probabilistic language modeling and Transformer attention

The first relevant lineage is neural language modeling. Neural probabilistic language models introduced the idea that discrete words can be mapped into continuous learned representations and used in a neural probability model over word sequences [1]. This move replaced sparse symbolic counting with differentiable geometry and remains one of the conceptual ancestors of modern language modeling.

The Transformer then made attention the dominant architectural mechanism for sequence transduction [2]. In the language used here, it constructs a dense relational substrate over token positions. This is precisely the point we formalize as complete-site presheaf dynamics: every token-token relation is made available before the model knows which relations are semantically or inferentially licensed.

2.2 Efficient sequence models and state-space alternatives

A large literature seeks to reduce the cost of full attention through sparse attention, linear attention, convolutional models, recurrent models, and structured state-space models. Mamba is particularly relevant because it identifies a weakness of earlier subquadratic models for discrete modalities and introduces selective state-space dynamics whose parameters depend on the input, allowing the model to selectively propagate or forget information along the sequence dimension [7]. Its reported linear scaling and strong language-modeling performance make it a natural source for the local state dynamics Φ_α in the proposed architecture.

However, state-space models by themselves do not specify a topology of typed language patches. They provide efficient sequential state evolution. The sheaf-state proposal adds a second layer: sparse compatibility-constrained gluing among local, instruction, memory, topic, retrieval, and workspace states. Thus the architecture is not simply an SSM replacement for attention; it is an attempt to organize SSM-like dynamics over a learned context site.

2.3 Sparse experts and low-bit morphisms

Sparse expert models show that not all parameters need to be active for every token. Switch Transformers use sparse activation to scale model capacity while limiting active computation through routing [6]. This supports our use of site-aware experts E_i , but the proposed router differs in its intended domain: it routes typed patch states rather than only token-level hidden states.

Low-bit language models, such as BitNet b1.58, support the hardware premise that the morphism category need not remain full high-precision $\mathbf{Vect}_{\mathbb{R}}$ at every stage [10]. In our formalism this motivates the use of **QuantVect**, a placeholder for quantized, sparse, or low-bit vector-like computation. Quantization is not the novelty; it is part of the hardware-aligned base category in which the gluing and local-state morphisms can be implemented.

2.4 Neural sheaf diffusion and sheaf neural networks

The closest mathematical relatives are sheaf neural networks. Neural Sheaf Diffusion equips graphs with cellular sheaves, assigning vector spaces and linear maps to graph elements, and shows that ordinary graph neural networks can be interpreted as operating over a trivial sheaf [3]. This is directly relevant to our diagnosis: a trivial or untyped relational geometry can limit the model’s ability to represent heterogeneous relations. Neural Sheaf Diffusion uses learned sheaves to address graph heterophily and oversmoothing, whereas the present work uses sheaf-like structure to diagnose and redesign language-context computation.

Connection-Laplacian sheaf neural networks extend this line by considering sheaf diffusion with connection Laplacians, emphasizing the difficulty and importance of choosing or learning the underlying sheaf structure [4]. Bayesian Sheaf Neural Networks further note that learning the sheaf from node features can introduce sensitivity and uncertainty, motivating Bayesian treatment of the sheaf itself [9]. These works are important because they show that learned sheaf structure is technically nontrivial. In our setting, this appears as the problem of learning the context topology J_{θ} and its gluing maps without destabilizing language modeling.

2.5 Sheaf attention and topological Transformers

The closest attention-specific precedent is sheaf attention. Attention-based Sheaf Neural Networks generalize attention mechanisms to operate over cellular sheaves and show that such constructions can help address heterophily and oversmoothing in graph settings [5]. This means that the phrase “attention as sheaf-like gluing” is not, by itself, novel.

Topological Transformer architectures also show that attention need not be restricted to flat token sequences or ordinary graphs. The Cellular Transformer generalizes graph-based Transformers to cell complexes, using incidence relations such as node-edge and edge-face relations to define topologically aware attention [8]. This is a crucial precedent for attention over nontrivial relational carriers.

The distinction is that these works usually start from a given graph, cellular sheaf, or cell complex. The present proposal treats autoregressive language context itself as the object to be topologized. The site is not an external data structure; it is the internal organization of context into local, syntactic, instructional, memory, retrieval, workspace, and output patches. The proposed architecture therefore moves from attention over a topological input object to topology as the learned organization of language-model context.

2.6 Categorical deep learning

Finally, the proposal belongs to the broader categorical deep-learning program, which uses functors, monoidal categories, optics, sheaves, and related structures to describe compositional learning architectures. This literature provides useful language, but it is not enough to justify a model. The categorical formalism is used here only insofar as it yields an architectural distinction: dense attention is modeled as presheaf dynamics over a complete token site, whereas the proposed model uses approximate sheaf dynamics over a sparse learned context site.

2.7 Novelty boundary

The novelty boundary is therefore narrow but clear. This work does not introduce neural language modeling, Transformer attention, sheaf neural networks, sheaf attention, state-space sequence modeling, sparse experts, or low-bit LLMs. Its contribution is to combine these directions under a specific architectural thesis:

LLM context should be represented as a learned sparse site of typed patches, and context integration should occur through recurrent local states plus sparse gluing morphisms rather than complete token-token attention.

In this sense, the paper is best read as a formal lower-complexity hypothesis for language modeling, not as an empirical claim of superiority.

3 Transformer Attention as Complete-Site Presheaf Dynamics

Let a tokenized sequence be

$$x = (x_1, \dots, x_n).$$

The Transformer associates to each token position i a vector state

$$h_i \in \mathbb{R}^d.$$

Let $\mathcal{C}_x$ be the small category whose objects are token positions. In dense self-attention, the effective relation graph on $\mathcal{C}_x$ is complete: each position i may receive information from each position j . A layer can therefore be interpreted as an operator on a presheaf

$$H : \mathcal{C}_x^{op} \rightarrow \mathbf{Vect}_{\mathbb{R}},$$

where $H(i) = h_i$.

The attention update is

$$h'_i = \sum_{j=1}^n \alpha_{ij} v_j,$$

with

$$\alpha_{ij} = \text{softmax}_j \left(\frac{q_i \cdot k_j}{\sqrt{d}} \right).$$

Categorically, this is a weighted aggregation of sections:

$$H'(i) = \int^{j \in \mathcal{C}_x} A_x(i, j) \otimes H(j),$$

where $A_x(i, j)$ is an input-dependent compatibility kernel.

This formulation reveals both the strength and the weakness of the Transformer. The strength is that every possible contextual morphism is available. The weakness is that availability, semantic relevance, inferential dependence, discourse relation, and memory requirement are collapsed into a single untyped compatibility coefficient.

4 Diagnosis: Overcomplete Relational Substrate

The coefficient α_{ij} does not distinguish relation types. It may encode local syntax, long-distance agreement, entity reference, instruction relevance, task dependence, topic continuity, or accidental statistical compatibility. These are structurally different relations, yet they enter the architecture through the same primitive operation: weighted token-token interaction.

Thus the architecture conflates

available relation

with

licensed semantic morphism.

The model does not merely learn relations; it learns them inside a complete graph where nearly all possible relations are offered as candidates. That gives expressive freedom, but it imposes quadratic relational complexity and high memory bandwidth. The proposed correction is to make the context topology itself part of the model.

5 Context as a Learned Site

Let

$$(\mathcal{C}, J)$$

be a context site. The objects of $\mathcal{C}$ are not only token positions but typed patches:

$$U_{\text{local}}, U_{\text{syntax}}, U_{\text{sentence}}, U_{\text{instruction}}, U_{\text{topic}}, U_{\text{memory}}, U_{\text{retrieval}}, U_{\text{workspace}}, U_{\text{output}}.$$

The Grothendieck topology J specifies admissible covers. A cover indicates which local patches may jointly determine a larger contextual state. A representation is a presheaf

$$F : \mathcal{C}^{op} \rightarrow \mathbf{QuantVect},$$

where **QuantVect** denotes a category of vector-like representations suitable for low-bit, sparse, or quantized computation.

The model is approximately sheaf-like when local sections agree on overlaps:

$$F(U) \approx \text{Eq} \left(\prod_{\alpha} F(U_{\alpha}) \rightrightarrows \prod_{\alpha, \beta} F(U_{\alpha} \cap U_{\beta}) \right).$$

This condition does not require exact symbolic consistency. It imposes a structural bias: global context should be assembled from compatible local states.

6 The Sheaf-State Language Model

Definition 1 (Sheaf-State Language Model). *A Sheaf-State Language Model is a tuple*

$$\mathfrak{M} = (\mathcal{C}, J, F, \Phi, G, E, R),$$

where $F : \mathcal{C}^{op} \rightarrow \mathbf{QuantVect}$ assigns states to context patches, $\Phi = \{\Phi_{\alpha}\}$ are local state-update morphisms, $G = \{g_{\beta\alpha}\}$ are sparse gluing morphisms, $E = \{E_i\}$ are site-aware expert morphisms, and R is the readout morphism.

Sheaf-State Language Model block schematic

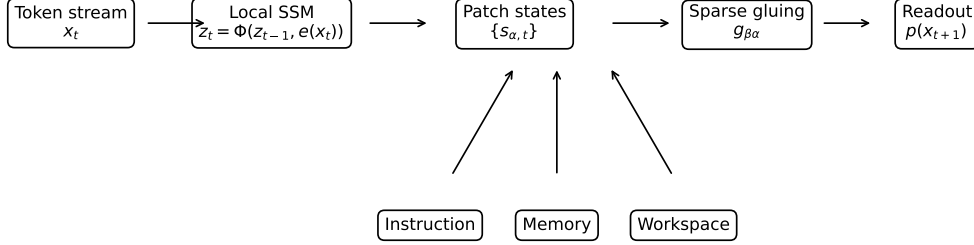


Figure 1: Schematic Sheaf-State Language Model block. A token stream is processed by local state-space dynamics, typed patch states are updated, sparse gluing exchanges information among compatible patches, and a readout state produces the next-token distribution.

At token time t , the model maintains a finite family of patch states

$$S_t = \{s_{\alpha,t}\}_{\alpha \in A}.$$

Local evolution is

$$\tilde{s}_{\alpha,t+1} = \Phi_{\alpha}(s_{\alpha,t}, e(x_t)).$$

Sparse gluing is

$$\hat{s}_{\alpha,t+1} = \tilde{s}_{\alpha,t+1} + \sum_{\beta \in N_t(\alpha)} g_{\beta\alpha}(\tilde{s}_{\beta,t+1}),$$

where $N_t(\alpha)$ is a small learned neighborhood of compatible patches. Expert refinement is

$$s_{\alpha,t+1} = E_{r(\alpha, \hat{s}_{\alpha,t+1})}(\hat{s}_{\alpha,t+1}).$$

The next-token distribution is

$$p(x_{t+1} \mid x_{\leq t}) = \text{softmax}(R(s_{\text{output},t+1})).$$

The training objective remains ordinary next-token prediction:

$$\mathcal{L}(\theta) = - \sum_t \log p_{\theta}(x_{t+1} \mid x_{\leq t}).$$

Thus the proposal does not change the basic statistical training signal. It changes the computational object on which language modeling is performed. Figure 1 summarizes the proposed block structure.

7 Complexity Analysis

Proposition 1 (Bounded gluing degree implies linear relational complexity). *Let (C_n, J_n) be a context site generated by a token sequence of length n . Suppose that each token contributes at most $|A|$ typed context patches and that each patch communicates through at most k gluing morphisms, where $|A|$ and k do not grow with n . Then the number of active relational morphisms scales as*

$$O(n|A|k).$$

If $|A|$ and k are bounded constants, relational complexity is $O(n)$. By contrast, dense self-attention over a complete token site computes $O(n^2)$ pairwise token-token compatibility scores.

Proof. There are at most $n|A|$ patch states. By assumption, each patch state communicates through at most k active gluing morphisms. Hence the total number of active gluing morphisms is at most $n|A|k$. If $|A|$ and k are independent of n , this is $O(n)$. Dense self-attention computes one compatibility score for each ordered or masked token-token pair, which is $O(n^2)$. $\square$

This proposition does not prove that a Sheaf-State Language Model has the same empirical performance as a Transformer. It proves a weaker and more precise claim: if useful context relations can be represented by a bounded-degree learned topology, then the relational substrate required for prediction scales linearly rather than quadratically with context length.

8 Synthetic Dependency Simulation

Because no full LLM implementation is claimed, the empirical component tests the structural hypothesis rather than language quality. We generate synthetic sequences with known sparse typed dependency graphs. Let

$$G_n^\star = (V_n, E_n^\star)$$

be the true dependency graph for a sequence of length n , with

$$|E_n^\star| = O(nk^\star).$$

The true graph contains three dependency classes: local token dependencies, long-range instruction/topic anchors, and entity-like recurrence links. We compare three architecture graphs:

$$G_{\text{dense}} = K_n,$$

$$G_{\text{local}} = \text{window graph of width } w,$$

and

$$G_{\text{sheaf}} = \text{typed patch graph with sparse covers.}$$

For each graph we compute dependency coverage

$$\text{coverage}(G) = \frac{|E_n^\star \cap E_G|}{|E_n^\star|}$$

and relational waste

$$\text{waste}(G) = \frac{|E_G \setminus E_n^\star|}{|E_G|}.$$

Dense attention should obtain full coverage but increasing waste when the true dependency graph is sparse. Local windows should reduce edges but miss some long-range dependencies. Sheaf-state

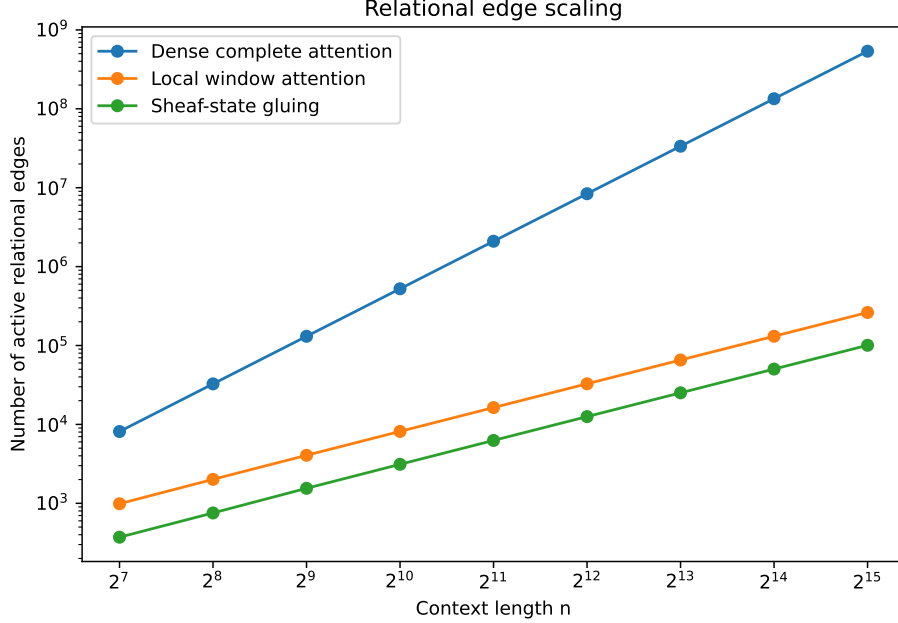


Figure 2: Relational edge scaling. Dense complete attention grows quadratically, whereas local window attention and sheaf-state gluing grow approximately linearly in the synthetic setting.

gluing should recover typed long-range dependencies when the cover structure matches the generating process. Figures 2–5 report the resulting edge scaling, dependency coverage, relational waste, and coverage–waste phase portrait.

At $n = 32768$, the synthetic true dependency graph contains 67908 sparse typed edges. Dense complete attention uses 536,854,528 model edges, obtains coverage 1.000, and has waste 0.999874. Local window attention uses 262,108 edges, obtains coverage 0.965, and has waste 0.749977. Sheaf-state gluing uses 100,653 edges, obtains coverage 1.000, and has waste 0.325326.

These results support only the structural claim. Dense attention is a safe but wasteful superset of the true dependency graph when the underlying dependency structure is sparse and typed. The simulation does not establish language-model quality, perplexity, instruction following, or benchmark performance.

9 Discussion

The proposed architecture should be read as a lower-complexity hypothesis class, not as a demonstrated replacement for Transformers. The contribution is the separation of three notions that dense attention collapses: availability, statistical compatibility, and licensed semantic or inferential dependence. In a Sheaf-State Language Model, these are redistributed across a learned topology J , local state morphisms Φ_α , sparse gluing maps $g_{\beta\alpha}$, and site-aware experts E_i .

This framework also clarifies how existing efficient-modeling directions can be combined without treating them as isolated engineering tricks. Selective state-space models supply local recurrent dynamics with linear sequence scaling [7]. Sparse experts reduce active parameter use by routing tokens or states to a subset of modules [6]. Low-bit language models suggest that the relevant morphism category need not be high-precision $\mathbf{Vect}_{\mathbb{R}}$ at every stage [10]. The sheaf-state proposal

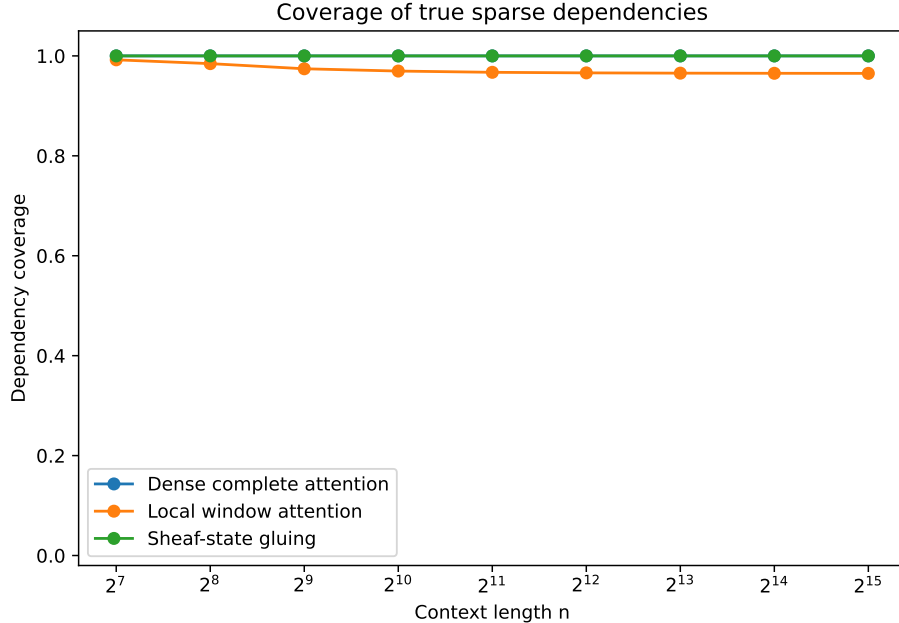


Figure 3: Dependency coverage. Dense attention recovers all true dependencies by construction. Local attention misses long-range typed dependencies. Sheaf-state gluing recovers the designed dependencies because its sparse covers include local, instruction/topic, and entity-like links.

organizes these ingredients around a single formal object: a sparse learned context site.

10 Limitations and Validation Protocol

The central limitation is that no trained language model is presented. Therefore, no claims are made about perplexity, downstream accuracy, instruction following, reasoning performance, or empirical superiority. The architecture may fail on arbitrary retrieval tasks unless it includes occasional global attention or an explicit retrieval patch. It may also underperform on in-context learning if the workspace patch lacks sufficient capacity.

A full validation protocol should include: (i) language modeling loss on small corpora, (ii) synthetic long-context retrieval and passkey tasks, (iii) structured dependency tasks where true relation graphs are known, (iv) latency and memory-bandwidth measurements, and (v) ablations over patch types, gluing degree, expert routing, and quantization level.

11 Conclusion

Dense self-attention can be interpreted as presheaf dynamics over a complete token site. This complete site is flexible, but structurally overcomplete. We proposed a Sheaf-State Language Model in which context is represented as a learned site of typed patches, local state-space dynamics maintain patch states, and sparse gluing morphisms exchange information only across compatible overlaps. The formal analysis shows linear relational complexity under bounded gluing degree. Synthetic dependency simulations show that complete attention obtains full coverage with asymptotically maximal waste, whereas sparse typed gluing can recover designed long-range dependencies with

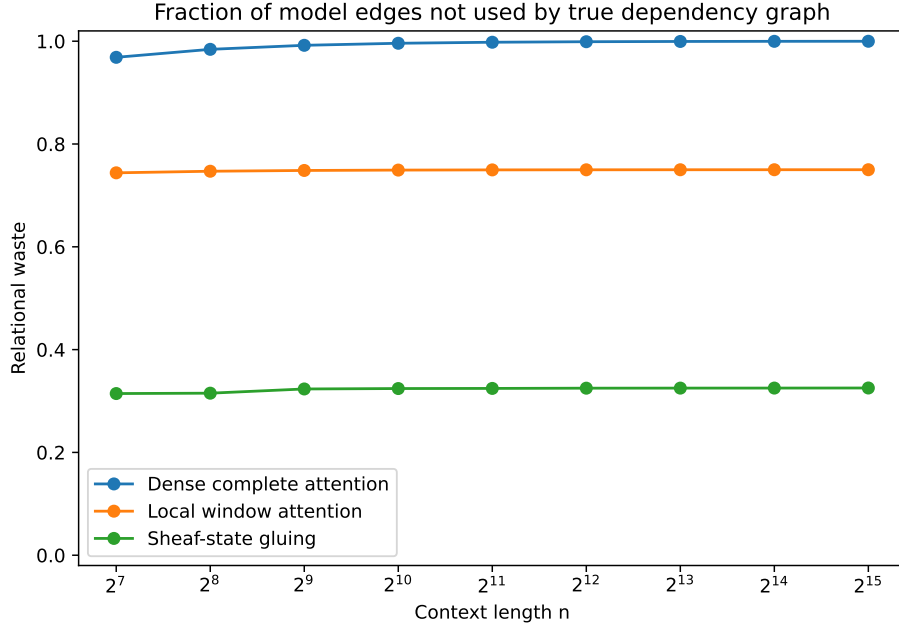


Figure 4: Relational waste. Dense attention becomes almost entirely wasteful as n grows under sparse true dependencies. Local attention and sheaf-state gluing use far fewer irrelevant edges.

far fewer active morphisms. The proposed research program is therefore not to approximate full attention, but to learn the right context topology.

Declarations

Funding. No external funding was received for this work.

Competing interests. The author declares no competing interests.

Data availability. The synthetic dependency data generated in this study are included in the submission package as `synthetic_dependency_results.csv`.

Code availability. The code used to generate the synthetic dependency simulation and figures is included in the submission package as `run_sheaf_state_simulation.py`.

Author contributions. Juan Segura conceived the study, developed the formal architecture, designed and analyzed the synthetic dependency simulation, prepared the figures, and wrote the manuscript.

References

- [1] Bengio, Y., Ducharme, R., Vincent, P., Janvin, C. (2003). A neural probabilistic language model. *Journal of Machine Learning Research*, 3, 1137–1155.

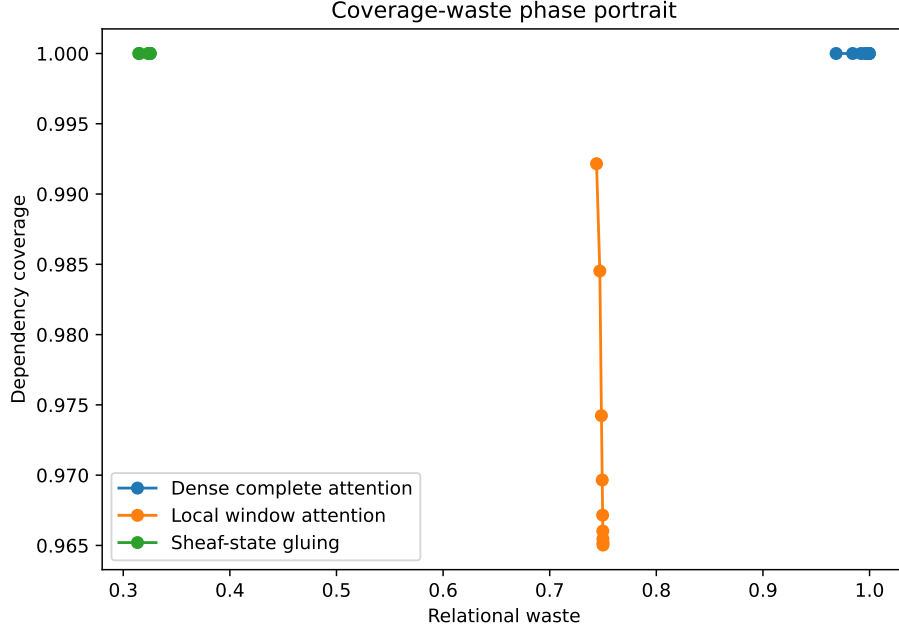


Figure 5: Coverage-waste phase portrait. Dense attention occupies the high-coverage/high-waste region. Local attention reduces waste but loses coverage. Sheaf-state gluing is designed to occupy the high-coverage/lower-waste region when the context topology captures the typed dependency structure.

- [2] Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A.N., Kaiser, L., Polosukhin, I. (2017). Attention is all you need. *Advances in Neural Information Processing Systems*.
- [3] Bodnar, C., Di Giovanni, F., Chamberlain, B.P., Liò, P., Bronstein, M.M. (2022). Neural sheaf diffusion: A topological perspective on heterophily and oversmoothing in GNNs. *Advances in Neural Information Processing Systems*.
- [4] Barbero, F., Bodnar, C., de Ocáriz Borde, H.S., Bronstein, M.M., Veličković, P., Liò, P. (2022). Sheaf neural networks with connection Laplacians. *arXiv preprint arXiv:2206.08702*.
- [5] Barbero, F. (2022). *Attention-based Sheaf Neural Networks*. Master’s thesis, University of Cambridge.
- [6] Fedus, W., Zoph, B., Shazeer, N. (2022). Switch Transformers: Scaling to trillion parameter models with simple and efficient sparsity. *Journal of Machine Learning Research*, 23(120), 1–39.
- [7] Gu, A., Dao, T. (2024). Mamba: Linear-time sequence modeling with selective state spaces. *International Conference on Learning Representations*.
- [8] Ballester, R., Rieck, B. (2024). Attending to topological spaces: The Cellular Transformer. *arXiv preprint arXiv:2405.14094*.
- [9] Gillespie, P. et al. (2024). Bayesian sheaf neural networks. *arXiv preprint arXiv:2410.09590*.

- [10] Ma, S., Wang, H., Ma, L., Wang, L., Wang, W., Huang, S., Dong, L., Wang, R., Xue, J., Wei, F. (2024). The era of 1-bit LLMs: All large language models are in 1.58 bits. *arXiv preprint arXiv:2402.17764*.