Investigating Representation Universality: Case Study on Genealogical Representations

Anonymous Author(s)
Affiliation
Address
email

Abstract

Motivated by interpretability and reliability, we investigate whether large language models (LLMs) deploy universal geometric structures to encode discrete, graph-structured knowledge. To this end, we present two complementary experimental evidence that might support universality of graph representations. First, on an in-context genealogy Q&A task, we train a cone probe to isolate a "tree-like" subspace in residual stream activations and use activation patching to verify its causal effect in answering related questions. We validate our findings across five different models. Second, we conduct model stitching experiments across models of diverse architectures and parameter counts (OPT, Pythia, Mistral, and LLaMA, 410 million to 8 billion parameters), quantifying representational alignment via relative degradation in the next-token prediction loss. Generally, we conclude that the lack of ground truth representations of graphs makes it challenging to study how LLMs represent them. Ultimately, improving our understanding of LLM representations could facilitate the development of more interpretable, robust, and controllable AI systems.

1 Introduction

Large Language Models (LLMs), despite being primarily trained for next-token predictions, have shown surprisingly robust reasoning capabilities (Bubeck et al., 2023; Anthropic, 2024; Team et al., 2023). However, despite recent progress, we still lack a clear understanding of how these models internally encode different kinds of knowledge. Improving such understanding could enable valuable progress relevant to transparency, interpretability, and safety; For example, (a) discovering and correcting inaccuracies to improve model reliability (Zhang et al., 2024a), (b) discovering and correcting biases (Chen et al., 2024), (c) revealing and removing dangerous knowledge (Zhang et al., 2024b), and (d) detecting deceptive behavior where models deliberately output information inconsistent with its knowledge (Marks and Tegmark, 2023).

Prior works have identified geometric structures of specific kinds of knowledge in LLMs and shown that these structures recur across many different models – evidence of representation universality. For example, Gurnee and Tegmark (2023) identified a linear subspace that captures spatio-temporal coordinates; Engels et al. (2024) discovered a circular manifold of calendar days and months' representations; and Kantamneni and Tegmark (2025) demonstrated a helical subspace of number representations. However, the question of how LLMs represent discrete, relational structures – such as nodes and edges in a knowledge graph – remains largely unexplored. In this paper, we ask:

Do LLMs exhibit representation universality when encoding graph-structured knowledge?

To investigate representation universality for discrete, graph-structured knowledge, we present two complementary experimental evidence:

Submitted to 39th Conference on Neural Information Processing Systems (NeurIPS 2025). Do not distribute.

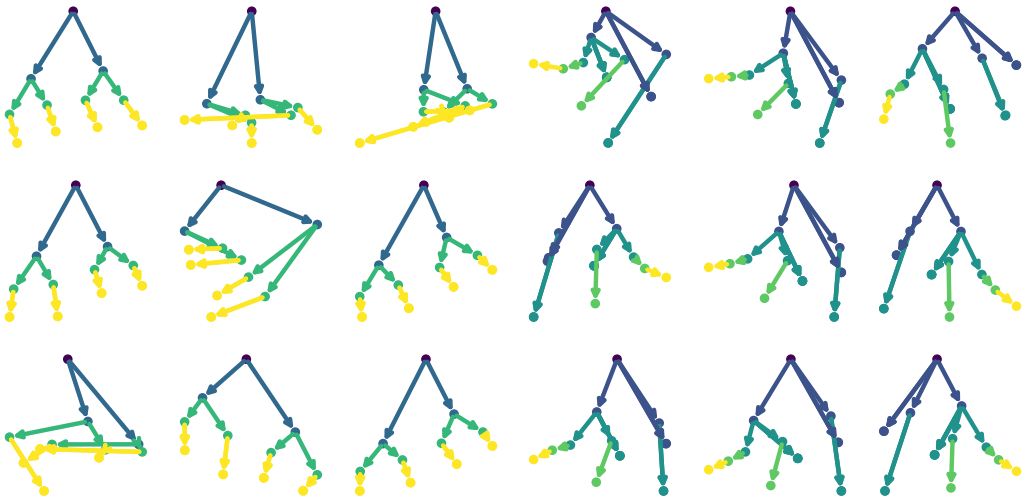

Figure 1: Visualization of the top two principal components of an MLP trained to learn the *descendant-of* relationship across nine different random seeds – for models trained on either (left) a fully balanced binary tree or a (right) randomly generated general tree consisting of 15 nodes. For clarity, we add arrows connecting direct parent–child pairs. Each plot is rotated so that the root node appears at the top of the panel. Across different seeds and tree structures, the learned representations consistently exhibit a geometric pattern that resembles a tree in discrete mathematics – a structure we define as *cone embedding* in the main text. Note that the models do not separate two sibling leaf nodes under the same parent. This is because all embeddings are initialized to zero, and the model receives no gradient signals to separate two sibling leaf nodes – they are equivalent nodes when it comes to determining the *descendant-of* relationship.

1. **Tree-structured subspace of Genealogy representations**: When representing *descendant-of* relationship, we identify that the optimal representation is a tree-like embedding that could be identified via cone probe. On an in-context genealogy Q&A task, we use a cone probe to isolate a tree-like subspace within the residual stream activations. We then use activation patching to verify the causality of this subspace. We validate our findings across five different models.

2. **Cross-model alignment via Model Stitching**: Since we lack a ground-truth representation for arbitrary graphs, we adopt a black-box model stitching approach to compare representations across different LLMs. We splice the early layers of one model onto the late layers of another via trainable linear adapter. Our experiments cover a diverse set of models – from OPT and Pythia to Mistral and LLaMA – ranging in size from 410 million to 8 billion parameters. By measuring the increase in next-token prediction loss relative to each model's baseline, we quantify representation alignment between different models.

Together, these experiments provide supporting evidence that LLMs may employ universal geometric structures to represent graphs. The rest of this paper is organized as follows: In Section 2, we formally describe the problem setting and introduce our hypothesis for the optimal representation of genealogical relationships. In Section 3, we investigate whether LLM representations exhibit geometric structure similar to the optimal representation we propose. Section 4 presents additional evidence for representational universality via LLM stitching experiments. We relate our approach to prior work in Section 5, and conclude our paper in Section 6.

2 Setup

Consider a general knowledge graph (KG) consisting of m binary relations $R^{(1)}, R^{(2)}, \cdots R^{(m)}$ between n objects (nodes) $x_1, ..., x_n$. Our task is to understand the representation that enables link prediction, the task of predicting the probability p_{ijk} that $R^{(i)}(x_j, x_k) = 1$. While most KG-learning algorithms in the literature embed both objects and relations (Cao et al., 2024), we instead embed only objects ($x_i \mapsto \mathbf{E}_i \equiv \mathbf{E}(x_i) \in \mathbb{R}^d$) and train a link predictor network $\mathbf{p}(\mathbf{E}_j, \mathbf{E}_k)$, which takes

two embedding vectors $\mathbf{E}_j, \mathbf{E}_k$ as an input, and outputs an m-dimensional vector $\mathbf{p}$ that represents link probability. This is to emulate the behavior of modern large language models, where only objects are embedded and relations are implicitly defined via weights. Our ultimate goal is to improve our understanding of representations that enable knowledge graph learning tasks in LLMs.

As a specific instance of this problem, consider a problem of learning *descendant-of* relationship in a tree. We claim that the optimal representation of this problem is a *cone* embedding, where j is a direct descendant of i iff $\mathbf{E}_j$ lies within a fixed cone emanating from $\mathbf{E}_i$.[1] We show that cone embedding is the optimal representation for this problem in Appendix A. Accordingly, we could define a score function which measures the probability that j is a direct descendant of i:

$$\mathbf{p}(\mathbf{E}_i, \mathbf{E}_j) = \sigma(\mathbf{E}_{i,1} - \mathbf{E}_{j,1})\sigma(\mathbf{E}_{i,0} - \mathbf{E}_{j,0}), \tag{1}$$

where σ is a sigmoid function and $\mathbf{E}_{i,n}$ denotes the $n-$th component of embedding $\mathbf{E}_i$. Since this score function is differentiable, we could also train a probe that measures how close a given embedding is to the cone embedding, which we refer to as the *cone probe*.

To test this hypothesis, we train a multi-layer perceptron (MLP) with a single hidden layer of width 50 to learn the *descendant-of* relationship on a tree consisting of 15 nodes. We do not use a test split, as our primary goal is to analyze the geometric structure of representations rather than to evaluate their generalization performance. The model embeds each object into a two-dimensional space, concatenates the resulting vectors, and passes them through the MLP to predict the probability that node j is a direct descendant of node i. We use the AdamW optimizer (Loshchilov and Hutter, 2017) with a learning rate of 10^{-3}, and train for up to 10^4 epochs, while applying early stopping if the loss does not improve for 30 consecutive epochs. We perform experiments on both a fully balanced binary tree and a randomly generated general tree.

Fig. 1 visualizes the top two principal components of the learned embeddings across nine different random seeds, both for the balanced tree and the general tree, with arrows added from each parent to its child for clarity. We observe that the learned representations indeed form *cone embeddings* – a geometric structure that closely resembles tree-like hierarchies in discrete mathematics. Another notable observation is that the model organizes the representations into a meaningful geometric structure, even though it could, in principle, simply memorize the training data and has no explicit incentive to learn a structured embedding. We hypothesize that this emergent structure is driven by the model's dimensionality constraint – specifically, the requirement to encode all relevant information within a two-dimensional space. This limitation effectively forces the model to arrange the objects into a coherent tree-like layout.

In the following section, we investigate whether genealogical representations in LLMs exhibit similar geometric structure. We will use the cone probe to identify relevant subspaces and perform causal interventions on them.

3 Genealogy Representations in LLMs

In the previous section, we observed that small models often encode genealogical relationships in a tree-like structure. This raises an interesting question: would LLMs represent genealogies in a similar way? To investigate, we design an in-context genealogy task as follows. We generate a full binary tree with 15 nodes and assign each node a name drawn at random (without replacement) from the 200 most popular male and female names from the birth year 2000, using the `pybabynames` package in Python (Balamuta, 2024). In the prompt, we first describe the family tree by listing all the children of every person on each line. We then ask questions of the form "Is *X* a direct descendant of *Y*?" to the LLM. We show an example of the full prompt in Appendix B. We evaluate our results over five different models, which are listed in Appendix C.

Fig. 2 shows the average F1 score on question-answering tasks about *descendant-of* relationships, averaged over five different name assignments on a tree. First, we found that the models are only able to answer these questions well when the lines, each of which lists the children of a specific person, are ordered based on the person's depth in the tree. When the orders are randomly shuffled, the model's performance significantly deteriorated. This is in accordance with the well-known reversal curse

[1] By *optimal*, we mean a representation that satisfies all the special properties of the relation, such as symmetricity and transitivity. We discuss optimal representations in more detail in Appendix A.

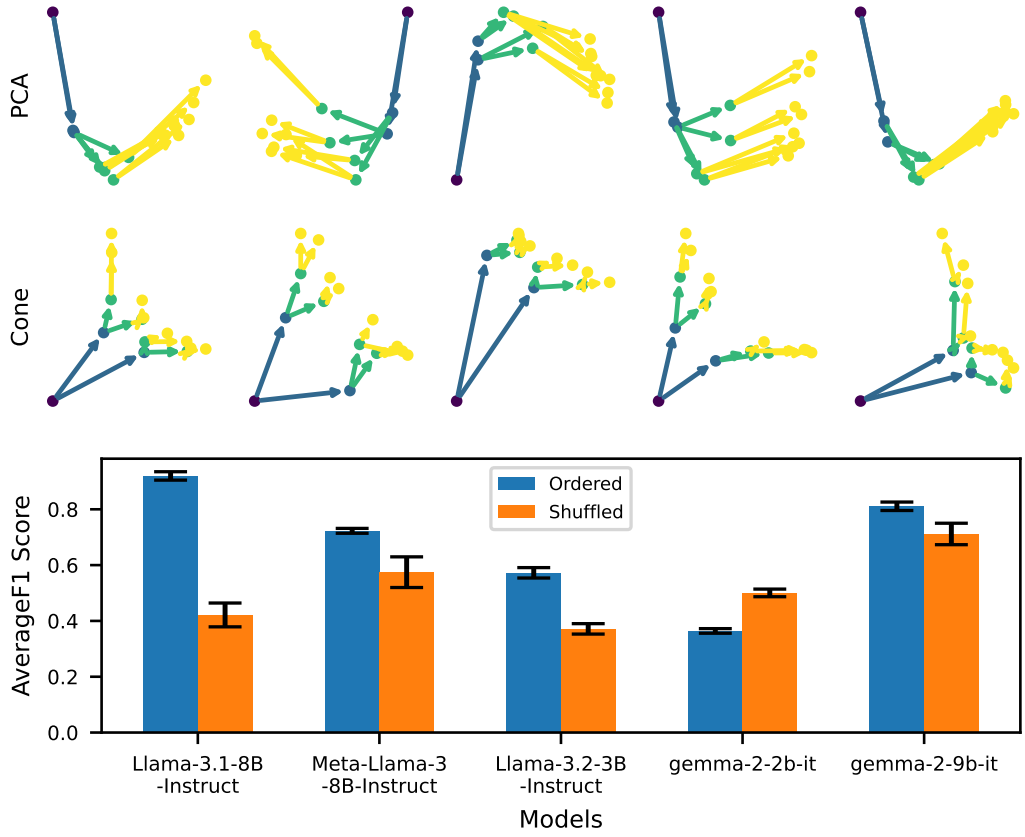

Figure 2: **Top**: Visualization of in-context genealogy-tree representations from LLaMA-3.1-8B-Instruct across five different random name assignments on a full binary tree of 15 nodes. We show the projection onto the first two principal components, and the Projection onto the cone-probe subspace. Nodes and edges are colored by their depth in the tree. We added arrows connecting direct parent-to-child links for visualization. **Bottom**: Average F1 score on question-answering tasks about *descendant-of* relationships, averaged over five different name assignments on a tree. These results suggest that the model may struggle with compositional generalization if the relevant facts are not provided in order.

(Berglund et al., 2023), where LLMs trained on "A is B" fail to learn "B is A". Hence, if we present "C is a child of D" first, and then "B is a child of C," the model may not be able to identify the reverse compositional relationship that "B is a child of D." Hence, we focus on studying the representations of the family tree when the graph descriptions are ordered based on people's depth in the tree.

To identify tree-like subspaces, we train a cone probe on the residual stream activations at the target token. To prevent overfitting, we first reduce each activation vector to 10 dimensions via PCA and then fit the cone probe in this lower dimensional space. We train a cone probe with AdamW optimizer with a learning rate 10^{-3} for 3000 epochs, while keeping the model that achieves the best F1 score on the original dataset. Fig. 2 visualizes the resulting 2D embeddings from PCA projection and cone projection across five different family trees (i.e. names are assigned to each node at random). We find that the PCA representations tends to be more degenerate (nodes at the same depth cluster tightly), whereas the cone probe yields a clearer, discrete "branching" structure that mirrors the underlying tree topology.

To verify the causal role of these subspaces, we conduct an intervention experiment. For each family tree, we sample 100 prompt pairs from five trees with different name assignments – a "clean" prompt (X, Y) and a corresponding "corrupted" prompt (X', Y) – constructed so their correct answers are opposite. We balance the set so that half of the clean prompts yield a positive (Yes) answer and the

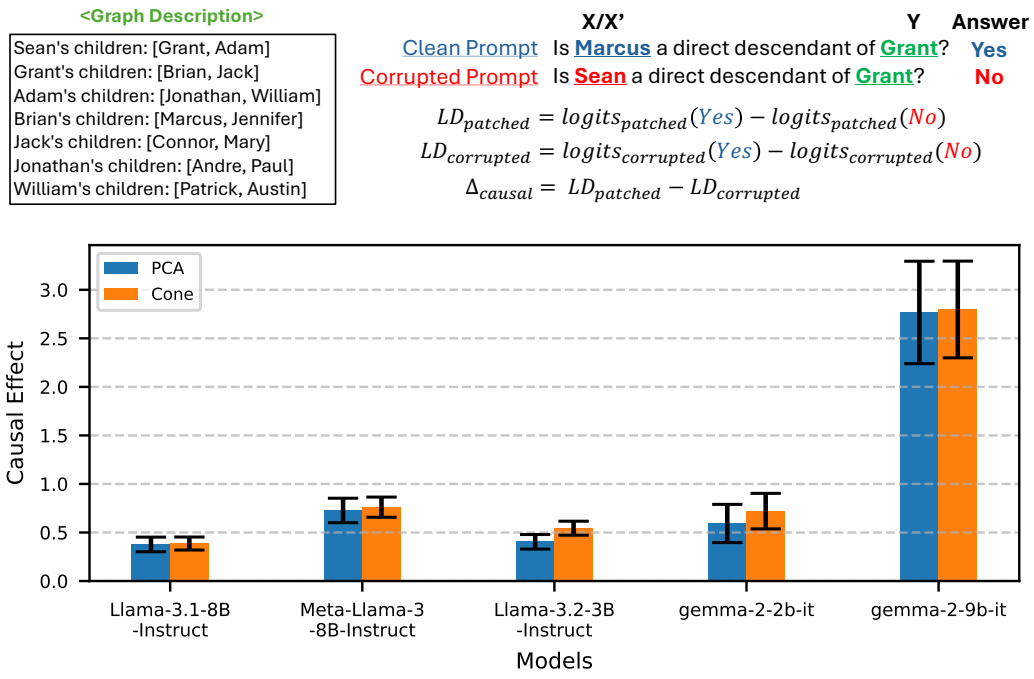

Figure 3: **Top**: Illustration of our intervention methodology. **Bottom**: Intervention results across five models. The histogram shows the causal effect of patching two subspaces of the residual stream activations at one-third model depth: (a) the subspace spanned by the top two principal components and (b) the cone subspace.

other half yield a negative (No) answer. For each pair, we run the model on the corrupt prompt, while patching its residual stream activations at layer l with those recorded from the clean run. We then quantify the causal effect of patching by comparing logit differences:

$$\Delta_{\text{causal}} = LD_{\text{patched}} - LD_{\text{corrupted}}, \tag{2}$$

where LD_x is the logit difference between the correct and incorrect tokens in run x. We compare three activation patching scenarios: (a) Patching the full layer, (b) Patching the top two principal components, and (c) Patching the cone subspace. More precisely, suppose $B \in \mathbb{R}^{d\times k}$ spans the subspace of interest, and define the orthogonal projection matrix

$$\mathbf{P} \;=\; B\,(B^\top B)^{-1}B^\top.$$

For any representation $\mathbf{x} \in \mathbb{R}^d$, the patched representation is given by

$$\mathbf{x}_{\text{patched}} = \mathbf{x}_{\text{corrupted}} - \mathbf{P}\mathbf{x}_{\text{corrupted}} + \mathbf{P}\mathbf{x}_{\text{clean}} \tag{3}$$

Intervention results are shown in Fig. 3 and Fig. 4. We find that representations in early to mid layers exhibit a stronger causal effect than those in later layers. Moreover, patching the cone-probe subspace alone produces a logit shift that is comparable to or larger than patching the top two principal components. Although full-layer patching yields an even larger effect – implying additional causally relevant directions beyond the cone subspace – our findings confirm that the cone subspace reliably emerges when models are asked to answer a question about a tree of relatively small size, and is at least as causally relevant as the subspace spanned by the top two principal components.

Limitations: First, we focus solely on the internal geometry and universality of LLM representations – without examining how these subspaces are actually leveraged by the model for answering questions, more well known as circuit analysis (Tigges et al., 2024). Second, our experiments use a relatively small binary tree consisting of 15 nodes on which models achieve near-perfect accuracy in answering

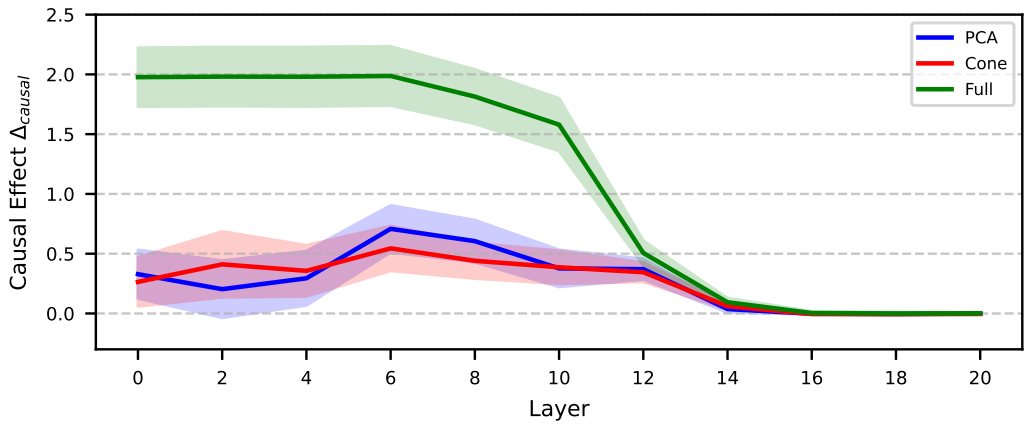

Figure 4: Intervention results for Llama-3.1-8B-Instruct across different layers. The plot shows the causal effect of patching two subspaces of the residual stream activations: (a) the subspace spanned by the top two principal components and (b) the cone subspace. Standard errors are indicated as a shaded region. Full represents patching the full activation at a specific layer.

related questions. Consequently, it remains unclear whether similar causal, tree-like subspaces emerge in larger or more complex genealogies, or if future, more capable models will encode genealogies in a similar manner. For instance, for a particular task of answering *descendant-of* questions, the ratio between positive and negative samples approaches zero as the tree size approaches infinity. Therefore, the model might just learn to say No for all questions while still getting accuracy larger than 99%. Hence, we would need a model that is good at what is known as the *needle-in-a-haystack* problem (Liu et al., 2023).

4 LLM Stitching Experiments

4.1 Model Stitching

Model Stitching (Lenc and Vedaldi, 2015; Bansal et al., 2021) is a method for probing the representation similarity between two different models by constructing a hybrid model that *stitches* the bottom layers of one model to the top layers of another model via trainable adapter layer. By measuring the performance drop of the stitched model relative to the original model, one could infer the degree of representation alignment between two different models. In this section, we apply this method to LLMs to study representation alignment between different LLMs.

Formally, the process of stitching two LLMs could be described as follows: Consider two LLMs

$$\mathbf{A} = \mathbf{U}^A \left(\prod_{i=0}^{n-1} \mathbf{H}_i \right) \mathbf{E}^A, \ \mathbf{B} = \mathbf{U}^B \left(\prod_{i=0}^{m-1} \mathbf{K}_i \right) \mathbf{E}^B, \tag{4}$$

where $\mathbf{H}_i, \mathbf{K}_i$ are decoder layers, $\mathbf{E}$ is the embedding layer, and $\mathbf{U}$ is the unembedding layer. The stitched model is given by

$$\mathbf{B} \circ \mathbf{A} = \mathbf{U}^B \left(\prod_{i=(m-l+1)}^{m-1} \mathbf{K}_i \right) S(\Lambda) \left(\prod_{i=0}^{k-1} \mathbf{H}_i \right) \mathbf{E}^A, \tag{5}$$

where we stitched the first k layers of $\mathbf{A}$ and the last l layers of $\mathbf{B}$. We then train a linear stitching layer $S(\Lambda)$ to minimize the next-token prediction cross-entropy loss:

$$\mathcal{L}(\Lambda) = \sum \log \left[\mathbb{P}(v_i | v_{i-1} \cdots v_0, \Lambda) \right]. \tag{6}$$

For stitching models with different tokenizers, v_i is the first token of the string $v_i v_{i+1} v_{i+2} \cdots$, tokenized by $\mathbf{B}$'s tokenizer. For our experiments, we stitch models from the OPT family (1.3B, 2.7B,

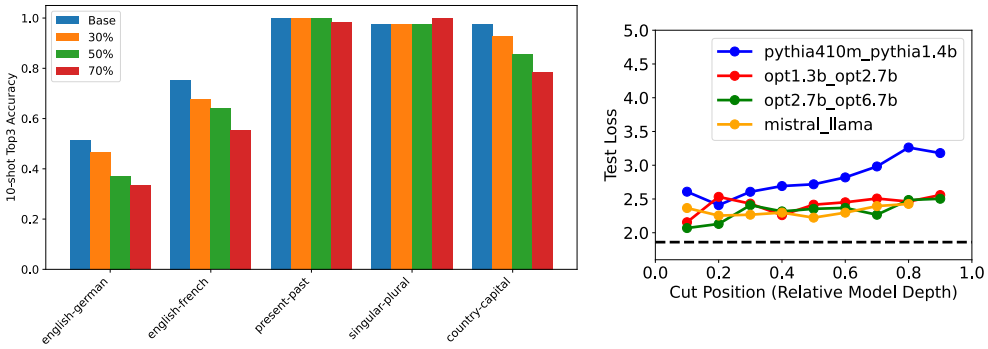

Figure 5: **Left**: In-context learning accuracy for models stitched between OPT-2.7B and OPT-6.7B. Base indicates OPT-6.7B, and $x\%$ indicate the embedding layer and first $x\%$ of the OPT-6.7B replaced by those of OPT-2.7B. **Right**: Test loss as a function of stitched position between two different models. The two models are cut at the same relative depth within each model. The black dashed line on the right figure indicates the average test loss of original models.

6.7B), Pythia family (410M, 1.4B, 2.8B), Mistral-7B-Instruct, and LLaMA-3.1-8B-Instruct. These models were chosen to cover a wide spectrum of model families and parameter scales. We trained the stitching layer for 10,000 steps with a linearly decaying learning rate starting at 10^{-3} with 100 warmup steps, and a weight decay of 10^{-4}. We used open-source models available in Huggingface, and used Huggingface datasets *monology/pile-uncopyrighted* and *monology/pile-test-val* for training and evaluating test loss. Each sample is truncated to 2048 tokens, and we report average test loss over 2000 randomly selected test samples.

4.2 Results

Fig. 5 presents the results of our LLM stitching experiments. Overall, we observe that representations from different models align more closely in early to mid layers than in later layers. Correspondingly, in-context learning performance declines as the stitching point moves to later relative depths.

We also evaluated stitching various layers of one model onto a fixed layer of another (Fig. 6 and Fig. 7). Test loss remains relatively low when connecting the embedding layer of one model to downstream layers of another, suggesting substantial representational transformations in the first few layers as token-level embeddings are converted into higher-level semantic concepts. While mid-layer representations between models are often compatible, stitching them into later layers yields higher loss – likely because those layers prioritize next-token prediction over forming semantic concepts. Interestingly, we can stitch a mid-layer of one model onto an early layer of another (e.g., layers 0–15 of Pythia-410M to layers 2–23 of Pythia-1.4B), implying that mid-layer activations still retain sufficient token-level information which could be "reset" to token-level representations.

These results corroborate the *Stages of Inferenece* hypothesis of Lad et al. (2024), which argues that LLMs process inputs through discrete phases – first constructing semantic representations in early-to-mid layers, then shifting to next-token prediction in later layers. Consequently, representations at equivalent relative depths across different models exhibit strong alignment, the concept known as representation universality (Huh et al., 2024).

Limitations: Despite its utility in quantifying representational alignment, our experiment has a few limitations. First, it assumes that representational alignment can be completely captured through a simple linear mapping; therefore, more complex or nonlinear representations, such as circular features in days of the month (Engels et al., 2024) or helical features in numbers (Kantamneni and Tegmark, 2025), may be classified as not equivalent. In order to circumvent this problem, one could consider adding a quadratic correction term to the adapter layer. Moreover, our experiments span only a limited set of LLM architectures and scales; therefore, it may not generalize to other models, such as multimodal models, that are not studied in this paper.

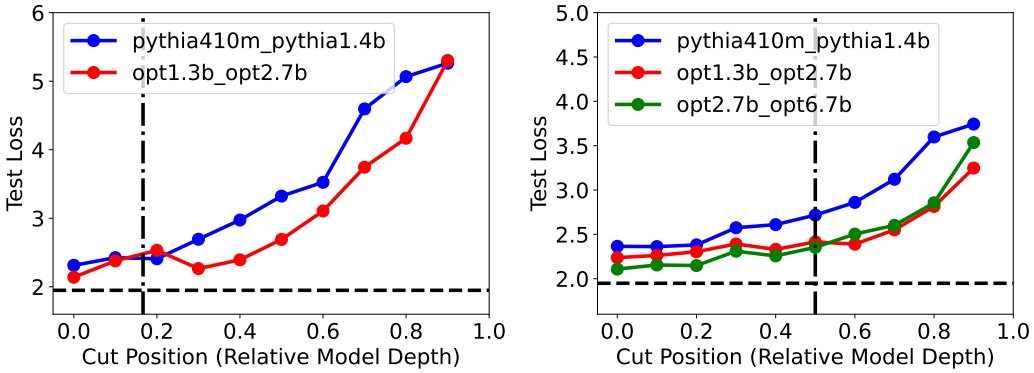

Figure 6: Test loss as a function of the stitch point x. We stitch the first $x\%$ of layers from the second model onto the first model at two fixed depths: one-sixth of its total depth (left) and one-half of its total depth (right). The vertical dashed line marks the relative depth where the first model is cut (one-sixth and one-half, respectively). The horizontal dashed line indicates the original models' average test loss.

5 Related Works

In light of the recent development of LLMs' capabilities, understanding the inner workings of Large Language Models have become increasingly important to ensure the safety and robustness of AI systems (Tegmark and Omohundro, 2023; Dalrymple et al., 2024).

Mechanistic Interpretability Neural Networks have demonstrated a surprising ability to generalize (Liu et al., 2021; Ye et al., 2021). Recently, there have been increasingly more efforts on trying to reverse engineer and interpret neural networks' internal operations (Zhang et al., 2021; Bereska and Gavves, 2024; Baek et al., 2024). Such methods include using structural probes and interventions at the level of entire representations (Hewitt and Manning, 2019; Pimentel et al., 2020), and studying neuron activations at the individual neuron level (Dalvi et al., 2019; Mu and Andreas, 2020). Our work is part of this broader effort in mechanistic interpretability; We aim to understand how large language models represent different types of knowledge.

Knowledge Representations in Language Models Early word-embedding models, including Word2Vec and GloVe, were found to encode semantic relationships as linear directions in their vector spaces (Drozd et al., 2016; Pennington et al., 2014; Ma and Zhang, 2015). More recently, several studies showed that LLMs are capable of forming conceptual representations in spatial, temporal, and color domains (Gurnee and Tegmark, 2023; Abdou et al., 2021; Li et al., 2021). Some studies focused primarily on examining the linearity of LLMs' feature representations (Gurnee and Tegmark, 2023; Hernandez et al., 2023). Several works found multi-dimensional representations of inputs such as lattices (Michaud et al., 2024) and circles (Liu et al., 2022; Engels et al., 2024), one-dimensional representations of high-level concepts and quantities in large language models (Gurnee and Tegmark, 2023; Marks and Tegmark, 2023; Heinzerling and Inui, 2024; Park et al., 2024b).

In particular, Park et al. (2024b) studied representations of word hierarchies. We examine representations of genealogical trees – another form of hierarchical data but are fundamentally different from word hierarchies because individuals in different generations do not possess inherent semantic relationships. Our findings suggest the existence of more multi-dimensional features, warranting further investigation. Our work is closely related to Park et al. (2024a), who study representations developed during in-context learning on a graph-tracing task. However, their analysis focuses on lattice and ring structures, which are inherently one-dimensional. In contrast, we aim to study in-context learning representations arising from data with more complex, hierarchical structures.

Our work is also closely related to traditional knowledge graph embedding models such as TransE (Wang et al., 2014), ComplexE (Trouillon et al., 2016), and TransR (Lin et al., 2015), which embed both entities and relations into a shared latent space and optimize a scoring function for link prediction. In contrast, our approach embeds only entities (objects), most closely mirroring how LLMs represent and process information.

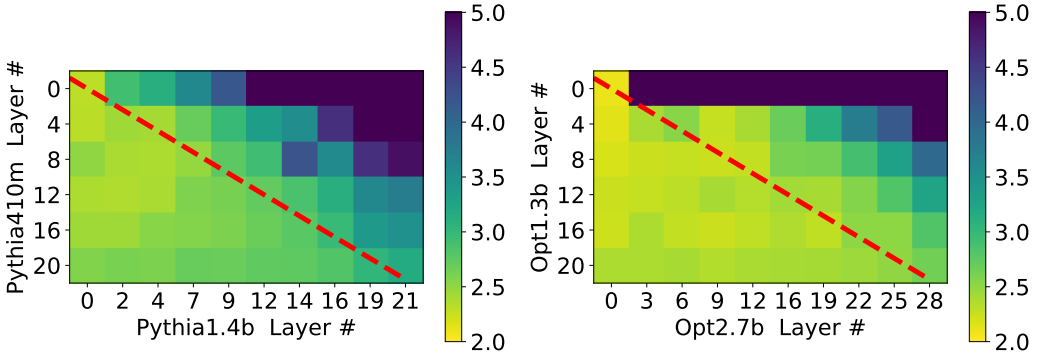

Figure 7: Test loss for different stitching configurations. Each point (i, j) indicates the loss of the model obtained by taking the first j layers of the y-axis model and the last $(L - j)$ layers of the x-axis model, where L is the total number of layers of the x-axis model. The red diagonal line marks the cases where both models are joined at the same relative depth.

Representation Alignment and Model Stitching There are active discussions in the literature about strengths and weaknesses of different representation alignment measures (Huh et al., 2024; Bansal et al., 2021; Sucholutsky et al., 2023). Several works have considered stitching to obtain better-performing models, such as stitching vision and language models for image and video captioning task (Li et al., 2019; Iashin and Rahtu, 2020; Shi et al., 2023), and stitching BERT and GPT for improved performance in look ahead section identification task (Jiang and Li, 2024). Some works have considered stitching toy transformers to understand the impact of activation functions on model's performance (Brown et al., 2023). Our work considers stitching LLMs to examine the hints of representation universality across different models.

6 Conclusion

We studied whether LLMs deploy universal geometric structures to encode graph-structured knowledge. We presented two complementary experimental evidence that supports universality of graph representations of LLMs. First, on an in-context genealogy Q&A task, we trained a cone probe to isolate a "tree-like" subspace in residual stream activations and utilized activation patching to verify its causal effect in answering related questions. Second, we conducted model stitching experiments across diverse architectures and parameter counts, and quantified representational alignment via relative degradation on next-token prediction loss. Generally, we conclude that the lack of ground truth representations of graphs makes it challenging to study how LLMs represent them. Ultimately, improving our understanding of LLM representations could facilitate the development of more interpretable, robust, and controllable AI systems.

Future Works: One could systematically investigate the optimal representations of more complex genealogical relationships – such as cousins, aunts, and uncles – and analyze whether LLMs encode these relations in a similar geometric manner. It would also be interesting to explore whether there exists a critical graph size beyond which such optimal representations begin to emerge. Our current study is limited to relatively small graphs, since model performance on genealogical question-answering tasks degrades significantly with increasing graph size. To address this, one could fine-tune existing LLMs or employ larger, more capable models to better understand the emergence of structured representations in larger graphs.

Another promising direction is to examine how LLMs estimate their uncertainty when reasoning over graph-structured data. We observe that LLMs rarely express full confidence in their answers to descendant-of questions, even for relatively small trees. Applying mechanistic interpretability techniques to study how uncertainty is represented could provide valuable insights into how LLMs process genealogical relationships in context.

References

Mostafa Abdou, Artur Kulmizev, Daniel Hershcovich, Stella Frank, Ellie Pavlick, and Anders Søgaard. Can language models encode perceptual structure without grounding? a case study in color. *arXiv preprint arXiv:2109.06129*, 2021.

Anthropic. The claude 3 model family: Opus, sonnet, haiku. Technical report, Anthropic, 2024.

David D Baek, Ziming Liu, and Max Tegmark. Geneft: Understanding statics and dynamics of model generalization via effective theory. *arXiv preprint arXiv:2402.05916*, 2024.

James Joseph Balamuta. pybabynames: Python port of the r data package *babynames*. https://pypi.org/project/pybabynames/, September 2024. Version 1.0.0; MIT License; accessed 2025-05-13.

Yamini Bansal, Preetum Nakkiran, and Boaz Barak. Revisiting model stitching to compare neural representations. *Advances in neural information processing systems*, 34:225–236, 2021.

Leonard Bereska and Efstratios Gavves. Mechanistic interpretability for ai safety–a review. *arXiv preprint arXiv:2404.14082*, 2024.

Lukas Berglund, Meg Tong, Max Kaufmann, Mikita Balesni, Asa Cooper Stickland, Tomasz Korbak, and Owain Evans. The reversal curse: Llms trained on" a is b" fail to learn" b is a". *arXiv preprint arXiv:2309.12288*, 2023.

Davis Brown, Charles Godfrey, Nicholas Konz, Jonathan Tu, and Henry Kvinge. Understanding the inner workings of language models through representation dissimilarity. *arXiv preprint arXiv:2310.14993*, 2023.

Sébastien Bubeck, Varun Chandrasekaran, Ronen Eldan, Johannes Gehrke, Eric Horvitz, Ece Kamar, Peter Lee, Yin Tat Lee, Yuanzhi Li, Scott Lundberg, et al. Sparks of artificial general intelligence: Early experiments with gpt-4. *arXiv preprint arXiv:2303.12712*, 2023.

Jiahang Cao, Jinyuan Fang, Zaiqiao Meng, and Shangsong Liang. Knowledge graph embedding: A survey from the perspective of representation spaces. *ACM Computing Surveys*, 56(6):1–42, 2024.

Ruizhe Chen, Yichen Li, Zikai Xiao, and Zuozhu Liu. Large language model bias mitigation from the perspective of knowledge editing. *arXiv preprint arXiv:2405.09341*, 2024.

David Dalrymple, Joar Skalse, Yoshua Bengio, Stuart Russell, Max Tegmark, Sanjit Seshia, Steve Omohundro, Christian Szegedy, Ben Goldhaber, Nora Ammann, et al. Towards guaranteed safe ai: A framework for ensuring robust and reliable ai systems. *arXiv preprint arXiv:2405.06624*, 2024.

Fahim Dalvi, Nadir Durrani, Hassan Sajjad, Yonatan Belinkov, Anthony Bau, and James Glass. What is one grain of sand in the desert? analyzing individual neurons in deep nlp models. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 33, pages 6309–6317, 2019.

Aleksandr Drozd, Anna Gladkova, and Satoshi Matsuoka. Word embeddings, analogies, and machine learning: Beyond king-man+ woman= queen. In *Proceedings of coling 2016, the 26th international conference on computational linguistics: Technical papers*, pages 3519–3530, 2016.

Joshua Engels, Isaac Liao, Eric J Michaud, Wes Gurnee, and Max Tegmark. Not all language model features are linear. *arXiv preprint arXiv:2405.14860*, 2024.

Google DeepMind. Gemma 2: Improving open language models at a practical size. *arXiv preprint arXiv:2408.00118*, 2024. URL https://arxiv.org/abs/2408.00118.

Wes Gurnee and Max Tegmark. Language models represent space and time. *arXiv preprint arXiv:2310.02207*, 2023.

Benjamin Heinzerling and Kentaro Inui. Monotonic representation of numeric properties in language models. *arXiv preprint arXiv:2403.10381*, 2024.

Evan Hernandez, Arnab Sen Sharma, Tal Haklay, Kevin Meng, Martin Wattenberg, Jacob Andreas, Yonatan Belinkov, and David Bau. Linearity of relation decoding in transformer language models. *arXiv preprint arXiv:2308.09124*, 2023.

John Hewitt and Christopher D Manning. A structural probe for finding syntax in word representations. 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 4129–4138, 2019.

Minyoung Huh, Brian Cheung, Tongzhou Wang, and Phillip Isola. The platonic representation hypothesis. *arXiv preprint arXiv:2405.07987*, 2024.

Vladimir Iashin and Esa Rahtu. Multi-modal dense video captioning. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition workshops*, pages 958–959, 2020.

Junlin Julian Jiang and Xin Li. Look ahead text understanding and llm stitching. In *Proceedings of the International AAAI Conference on Web and Social Media*, volume 18, pages 751–760, 2024.

Subhash Kantamneni and Max Tegmark. Language models use trigonometry to do addition. *arXiv preprint arXiv:2502.00873*, 2025.

Vedang Lad, Wes Gurnee, and Max Tegmark. The remarkable robustness of llms: Stages of inference? *arXiv preprint arXiv:2406.19384*, 2024.

Karel Lenc and Andrea Vedaldi. Understanding image representations by measuring their equivariance and equivalence. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 991–999, 2015.

Belinda Z Li, Maxwell Nye, and Jacob Andreas. Implicit representations of meaning in neural language models. *arXiv preprint arXiv:2106.00737*, 2021.

Sheng Li, Zhiqiang Tao, Kang Li, and Yun Fu. Visual to text: Survey of image and video captioning. *IEEE Transactions on Emerging Topics in Computational Intelligence*, 3(4):297–312, 2019.

Yankai Lin, Zhiyuan Liu, Maosong Sun, Yang Liu, and Xuan Zhu. Learning entity and relation embeddings for knowledge graph completion. In *Proceedings of the AAAI conference on artificial intelligence*, volume 29, 2015.

Jiashuo Liu, Zheyan Shen, Yue He, Xingxuan Zhang, Renzhe Xu, Han Yu, and Peng Cui. Towards out-of-distribution generalization: A survey. *arXiv preprint arXiv:2108.13624*, 2021.

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

Ziming Liu, Ouail Kitouni, Niklas S Nolte, Eric Michaud, Max Tegmark, and Mike Williams. Towards understanding grokking: An effective theory of representation learning. *Advances in Neural Information Processing Systems*, 35:34651–34663, 2022.

Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization. *arXiv preprint arXiv:1711.05101*, 2017.

Long Ma and Yanqing Zhang. Using word2vec to process big text data. In *2015 IEEE International Conference on Big Data (Big Data)*, pages 2895–2897. IEEE, 2015.

Samuel Marks and Max Tegmark. The geometry of truth: Emergent linear structure in large language model representations of true/false datasets. *arXiv preprint arXiv:2310.06824*, 2023.

Eric J Michaud, Isaac Liao, Vedang Lad, Ziming Liu, Anish Mudide, Chloe Loughridge, Zifan Carl Guo, Tara Rezaei Kheirkhah, Mateja Vukelić, and Max Tegmark. Opening the ai black box: program synthesis via mechanistic interpretability. *arXiv preprint arXiv:2402.05110*, 2024.

Jesse Mu and Jacob Andreas. Compositional explanations of neurons. *Advances in Neural Information Processing Systems*, 33:17153–17163, 2020.

Core Francisco Park, Andrew Lee, Ekdeep Singh Lubana, Yongyi Yang, Maya Okawa, Kento Nishi, Martin Wattenberg, and Hidenori Tanaka. Iclr: In-context learning of representations. *arXiv preprint arXiv:2501.00070*, 2024a.

Kiho Park, Yo Joong Choe, Yibo Jiang, and Victor Veitch. The geometry of categorical and hierarchical concepts in large language models. *arXiv preprint arXiv:2406.01506*, 2024b.

Jeffrey Pennington, Richard Socher, and Christopher D Manning. Glove: Global vectors for word representation. In *Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP)*, pages 1532–1543, 2014.

Tiago Pimentel, Josef Valvoda, Rowan Hall Maudslay, Ran Zmigrod, Adina Williams, and Ryan Cotterell. Information-theoretic probing for linguistic structure. *arXiv preprint arXiv:2004.03061*, 2020.

Yaya Shi, Haiyang Xu, Chunfeng Yuan, Bing Li, Weiming Hu, and Zheng-Jun Zha. Learning video-text aligned representations for video captioning. *ACM Transactions on Multimedia Computing, Communications and Applications*, 19(2):1–21, 2023.

Ilia Sucholutsky, Lukas Muttenthaler, Adrian Weller, Andi Peng, Andreea Bobu, Been Kim, Bradley C Love, Erin Grant, Jascha Achterberg, Joshua B Tenenbaum, et al. Getting aligned on representational alignment. *arXiv preprint arXiv:2310.13018*, 2023.

Gemini Team, Rohan Anil, Sebastian Borgeaud, Yonghui Wu, Jean-Baptiste Alayrac, Jiahui Yu, Radu Soricut, Johan Schalkwyk, Andrew M Dai, Anja Hauth, et al. Gemini: a family of highly capable multimodal models. *arXiv preprint arXiv:2312.11805*, 2023.

Max Tegmark and Steve Omohundro. Provably safe systems: the only path to controllable agi. *arXiv preprint arXiv:2309.01933*, 2023.

Curt Tigges, Michael Hanna, Qinan Yu, and Stella Biderman. Llm circuit analyses are consistent across training and scale. *arXiv preprint arXiv:2407.10827*, 2024.

Hugo Touvron et al. The llama 3 herd of models. *arXiv preprint arXiv:2407.21783*, 2024. URL https://arxiv.org/abs/2407.21783.

Théo Trouillon, Johannes Welbl, Sebastian Riedel, Éric Gaussier, and Guillaume Bouchard. Complex embeddings for simple link prediction. In *International conference on machine learning*, pages 2071–2080. PMLR, 2016.

Zhen Wang, Jianwen Zhang, Jianlin Feng, and Zheng Chen. Knowledge graph embedding by translating on hyperplanes. In *Proceedings of the AAAI conference on artificial intelligence*, volume 28, 2014.

Haotian Ye, Chuanlong Xie, Tianle Cai, Ruichen Li, Zhenguo Li, and Liwei Wang. Towards a theoretical framework of out-of-distribution generalization. *Advances in Neural Information Processing Systems*, 34:23519–23531, 2021.

Xiaoying Zhang, Baolin Peng, Ye Tian, Jingyan Zhou, Lifeng Jin, Linfeng Song, Haitao Mi, and Helen Meng. Self-alignment for factuality: Mitigating hallucinations in llms via self-evaluation. *arXiv preprint arXiv:2402.09267*, 2024a.

Yihao Zhang, Zeming Wei, Jun Sun, and Meng Sun. Adversarial representation engineering: A general model editing framework for large language models. *arXiv preprint arXiv:2404.13752*, 2024b.

Yu Zhang, Peter Tiňo, Aleš Leonardis, and Ke Tang. A survey on neural network interpretability. *IEEE Transactions on Emerging Topics in Computational Intelligence*, 5(5):726–742, 2021.

A Optimal Representation in Knowledge Graph Learning

We define *optimal* representation as those that satisfiy all the special properties of the relation. Such properties include

- Symmetricity: $\forall x_1, x_2 : R(x_1, x_2) \implies R(x_1, x_2)$
- Reflexivity: $\forall x_1 : R(x_1, x_1) = 1$
- Transitivity: $\forall i, j, k : R(x_i, x_j) \wedge R(x_j, x_k) \implies R(x_i, x_k)$
- Meta-transitivity: $\forall i, j, k : R^{(1)}(x_i, x_j) \wedge R^{(1)}(x_j, x_k) \implies R^{(2)}(x_i, x_k)$

As an example, we prove that *cone* embedding in the main text is an optimal representation of the *descendant-of* relationship.

Proof. Our predictor function for cone probe is given by $\mathbf{p}(E_i, E_j) = H(E_{i0} - E_{j0})H(E_{i1} - E_{j1})$ where H is the heaviside step function ($H(x) = 1$ if $x > 0$, vanishing otherwise). We show that $\mathbf{p}$ satisfies transitivity, *i.e.*if i is a descendant of j, and j is a descendant of k, then i is a descendant of k:

Suppose $\mathbf{p}(E_i, E_j) = \mathbf{p}(E_j, E_k) = 1$. By definition of the cone probe,

$$\mathbf{p}(E_i, E_j) = 1 \iff E_{i0} > E_{j0} \wedge E_{i1} > E_{j1}, \quad \mathbf{p}(E_j, E_k) = 1 \iff E_{j0} > E_{k0} \wedge E_{j1} > E_{k1}.$$

Chaining these inequalities gives

$$E_{i0} > E_{k0} \quad \text{and} \quad E_{i1} > E_{k1},$$

and hence $\mathbf{p}(E_i, E_k) = H(E_{i0} - E_{k0})\, H(E_{i1} - E_{k1}) = 1$. □

B Full Prompt Example

```
Below is an instruction that describes a task, paired with an input
    ↪ that provides further context. Write a response that
    ↪ appropriately completes the request.

### Instruction:
Answer a question about the family tree relationship based on the
    ↪ given data. If it's a yes/no question, answer with only one
    ↪ word: 'Yes' or 'No.' If it's a 'who' question, answer with the
    ↪ person's name(s).

### Input:
Family Tree:
Emily's children: [Scott, Jordan]
Scott's children: [Marco, William]
Jordan's children: [Charles, Hunter]
Marco's children: [Luke, Jose]
William's children: [Jessica, Crystal]
Charles's children: [Alan, Joseph]
Hunter's children: [Laura, Grace]

Question: Is Grace a direct descendant of Laura?

### Response:
```

C List of Models

Model Name	Citation
meta-llama/llama-3.1-8b-instruct	Touvron et al. (2024)
meta-llama/Meta-Llama-3-8B-Instruct	Touvron et al. (2024)
meta-llama/Llama-3.2-3B-Instruct	Touvron et al. (2024)
google/gemma-2-2b-it	Google DeepMind (2024)
google/gemma-2-9b-it	Google DeepMind (2024)

Table 1: List of Models used in our experiments.

NeurIPS Paper Checklist

1. **Claims**

   Question: Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope?

   Answer: [Yes]

   Justification: All of our claims in the abstract and intro are reflected in the paper; we list out the exact section where each contribution is shown in our list of contributions in the intro.

   Guidelines:

   - The answer NA means that the abstract and introduction do not include the claims made in the paper.
   - The abstract and/or introduction should clearly state the claims made, including the contributions made in the paper and important assumptions and limitations. A No or NA answer to this question will not be perceived well by the reviewers.
   - The claims made should match theoretical and experimental results, and reflect how much the results can be expected to generalize to other settings.
   - It is fine to include aspirational goals as motivation as long as it is clear that these goals are not attained by the paper.

2. **Limitations**

   Question: Does the paper discuss the limitations of the work performed by the authors?

   Answer: [Yes]

   Justification: We discuss the limitations when we discuss results in each section, and we discuss overall limitations in the conclusion.

   Guidelines:

   - The answer NA means that the paper has no limitation while the answer No means that the paper has limitations, but those are not discussed in the paper.
   - The authors are encouraged to create a separate "Limitations" section in their paper.
   - The paper should point out any strong assumptions and how robust the results are to violations of these assumptions (e.g., independence assumptions, noiseless settings, model well-specification, asymptotic approximations only holding locally). The authors should reflect on how these assumptions might be violated in practice and what the implications would be.
   - The authors should reflect on the scope of the claims made, e.g., if the approach was only tested on a few datasets or with a few runs. In general, empirical results often depend on implicit assumptions, which should be articulated.
   - The authors should reflect on the factors that influence the performance of the approach. For example, a facial recognition algorithm may perform poorly when image resolution is low or images are taken in low lighting. Or a speech-to-text system might not be used reliably to provide closed captions for online lectures because it fails to handle technical jargon.
   - The authors should discuss the computational efficiency of the proposed algorithms and how they scale with dataset size.

- If applicable, the authors should discuss possible limitations of their approach to address problems of privacy and fairness.
- While the authors might fear that complete honesty about limitations might be used by reviewers as grounds for rejection, a worse outcome might be that reviewers discover limitations that aren't acknowledged in the paper. The authors should use their best judgment and recognize that individual actions in favor of transparency play an important role in developing norms that preserve the integrity of the community. Reviewers will be specifically instructed to not penalize honesty concerning limitations.

3. **Theory assumptions and proofs**

   Question: For each theoretical result, does the paper provide the full set of assumptions and a complete (and correct) proof?

   Answer: [Yes]

   Justification: We provide proofs that the cone representation is optimal for representing descendant-of relationship in Appendix A.

   Guidelines:

   - The answer NA means that the paper does not include theoretical results.
   - All the theorems, formulas, and proofs in the paper should be numbered and cross-referenced.
   - All assumptions should be clearly stated or referenced in the statement of any theorems.
   - The proofs can either appear in the main paper or the supplemental material, but if they appear in the supplemental material, the authors are encouraged to provide a short proof sketch to provide intuition.
   - Inversely, any informal proof provided in the core of the paper should be complemented by formal proofs provided in appendix or supplemental material.
   - Theorems and Lemmas that the proof relies upon should be properly referenced.

4. **Experimental result reproducibility**

   Question: Does the paper fully disclose all the information needed to reproduce the main experimental results of the paper to the extent that it affects the main claims and/or conclusions of the paper (regardless of whether the code and data are provided or not)?

   Answer: [Yes]

   Justification: We include sufficient experiment details to reproduce our experiments, and we include our code for the full details, and we include our code for the full details.

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - If the paper includes experiments, a No answer to this question will not be perceived well by the reviewers: Making the paper reproducible is important, regardless of whether the code and data are provided or not.
   - If the contribution is a dataset and/or model, the authors should describe the steps taken to make their results reproducible or verifiable.
   - Depending on the contribution, reproducibility can be accomplished in various ways. For example, if the contribution is a novel architecture, describing the architecture fully might suffice, or if the contribution is a specific model and empirical evaluation, it may be necessary to either make it possible for others to replicate the model with the same dataset, or provide access to the model. In general. releasing code and data is often one good way to accomplish this, but reproducibility can also be provided via detailed instructions for how to replicate the results, access to a hosted model (e.g., in the case of a large language model), releasing of a model checkpoint, or other means that are appropriate to the research performed.
   - While NeurIPS does not require releasing code, the conference does require all submissions to provide some reasonable avenue for reproducibility, which may depend on the nature of the contribution. For example
     (a) If the contribution is primarily a new algorithm, the paper should make it clear how to reproduce that algorithm.

(b) If the contribution is primarily a new model architecture, the paper should describe the architecture clearly and fully.
(c) If the contribution is a new model (e.g., a large language model), then there should either be a way to access this model for reproducing the results or a way to reproduce the model (e.g., with an open-source dataset or instructions for how to construct the dataset).
(d) We recognize that reproducibility may be tricky in some cases, in which case authors are welcome to describe the particular way they provide for reproducibility. In the case of closed-source models, it may be that access to the model is limited in some way (e.g., to registered users), but it should be possible for other researchers to have some path to reproducing or verifying the results.

5. **Open access to data and code**

   Question: Does the paper provide open access to the data and code, with sufficient instructions to faithfully reproduce the main experimental results, as described in supplemental material?

   Answer: [Yes]

   Justification: Our anonymous code is available at https://anonymous.4open.science/r/llm-tree-6351.

   Guidelines:

   - The answer NA means that paper does not include experiments requiring code.
   - Please see the NeurIPS code and data submission guidelines (https://nips.cc/public/guides/CodeSubmissionPolicy) for more details.
   - While we encourage the release of code and data, we understand that this might not be possible, so "No" is an acceptable answer. Papers cannot be rejected simply for not including code, unless this is central to the contribution (e.g., for a new open-source benchmark).
   - The instructions should contain the exact command and environment needed to run to reproduce the results. See the NeurIPS code and data submission guidelines (https://nips.cc/public/guides/CodeSubmissionPolicy) for more details.
   - The authors should provide instructions on data access and preparation, including how to access the raw data, preprocessed data, intermediate data, and generated data, etc.
   - The authors should provide scripts to reproduce all experimental results for the new proposed method and baselines. If only a subset of experiments are reproducible, they should state which ones are omitted from the script and why.
   - At submission time, to preserve anonymity, the authors should release anonymized versions (if applicable).
   - Providing as much information as possible in supplemental material (appended to the paper) is recommended, but including URLs to data and code is permitted.

6. **Experimental setting/details**

   Question: Does the paper specify all the training and test details (e.g., data splits, hyperparameters, how they were chosen, type of optimizer, etc.) necessary to understand the results?

   Answer: [Yes]

   Justification: We specify all experiment details for each of our experiment.

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - The experimental setting should be presented in the core of the paper to a level of detail that is necessary to appreciate the results and make sense of them.
   - The full details can be provided either with the code, in appendix, or as supplemental material.

7. **Experiment statistical significance**

   Question: Does the paper report error bars suitably and correctly defined or other appropriate information about the statistical significance of the experiments?

Answer: [Yes]

Justification: We report 1-sigma error bars for the experiments.

Guidelines:

- The answer NA means that the paper does not include experiments.
- The authors should answer "Yes" if the results are accompanied by error bars, confidence intervals, or statistical significance tests, at least for the experiments that support the main claims of the paper.
- The factors of variability that the error bars are capturing should be clearly stated (for example, train/test split, initialization, random drawing of some parameter, or overall run with given experimental conditions).
- The method for calculating the error bars should be explained (closed form formula, call to a library function, bootstrap, etc.)
- The assumptions made should be given (e.g., Normally distributed errors).
- It should be clear whether the error bar is the standard deviation or the standard error of the mean.
- It is OK to report 1-sigma error bars, but one should state it. The authors should preferably report a 2-sigma error bar than state that they have a 96% CI, if the hypothesis of Normality of errors is not verified.
- For asymmetric distributions, the authors should be careful not to show in tables or figures symmetric error bars that would yield results that are out of range (e.g. negative error rates).
- If error bars are reported in tables or plots, The authors should explain in the text how they were calculated and reference the corresponding figures or tables in the text.

8. **Experiments compute resources**

   Question: For each experiment, does the paper provide sufficient information on the computer resources (type of compute workers, memory, time of execution) needed to reproduce the experiments?

   Answer: [Yes]

   Justification: All of our experiments could be reproduced with one A100 GPU.

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - The paper should indicate the type of compute workers CPU or GPU, internal cluster, or cloud provider, including relevant memory and storage.
   - The paper should provide the amount of compute required for each of the individual experimental runs as well as estimate the total compute.
   - The paper should disclose whether the full research project required more compute than the experiments reported in the paper (e.g., preliminary or failed experiments that didn't make it into the paper).

9. **Code of ethics**

   Question: Does the research conducted in the paper conform, in every respect, with the NeurIPS Code of Ethics `https://neurips.cc/public/EthicsGuidelines`?

   Answer: [Yes]

   Justification: Our work follows the NeurIPS Code of Ethics.

   Guidelines:

   - The answer NA means that the authors have not reviewed the NeurIPS Code of Ethics.
   - If the authors answer No, they should explain the special circumstances that require a deviation from the Code of Ethics.
   - The authors should make sure to preserve anonymity (e.g., if there is a special consideration due to laws or regulations in their jurisdiction).

10. **Broader impacts**

Question: Does the paper discuss both potential positive societal impacts and negative societal impacts of the work performed?

Answer: [Yes]

Justification: We discuss potential impacts of our work in the conclusion. We do not foresee any negative implications of our work.

Guidelines:

- The answer NA means that there is no societal impact of the work performed.
- If the authors answer NA or No, they should explain why their work has no societal impact or why the paper does not address societal impact.
- Examples of negative societal impacts include potential malicious or unintended uses (e.g., disinformation, generating fake profiles, surveillance), fairness considerations (e.g., deployment of technologies that could make decisions that unfairly impact specific groups), privacy considerations, and security considerations.
- The conference expects that many papers will be foundational research and not tied to particular applications, let alone deployments. However, if there is a direct path to any negative applications, the authors should point it out. For example, it is legitimate to point out that an improvement in the quality of generative models could be used to generate deepfakes for disinformation. On the other hand, it is not needed to point out that a generic algorithm for optimizing neural networks could enable people to train models that generate Deepfakes faster.
- The authors should consider possible harms that could arise when the technology is being used as intended and functioning correctly, harms that could arise when the technology is being used as intended but gives incorrect results, and harms following from (intentional or unintentional) misuse of the technology.
- If there are negative societal impacts, the authors could also discuss possible mitigation strategies (e.g., gated release of models, providing defenses in addition to attacks, mechanisms for monitoring misuse, mechanisms to monitor how a system learns from feedback over time, improving the efficiency and accessibility of ML).

11. **Safeguards**

Question: Does the paper describe safeguards that have been put in place for responsible release of data or models that have a high risk for misuse (e.g., pretrained language models, image generators, or scraped datasets)?

Answer: [NA]

Justification: We do not release any assets.

Guidelines:

- The answer NA means that the paper poses no such risks.
- Released models that have a high risk for misuse or dual-use should be released with necessary safeguards to allow for controlled use of the model, for example by requiring that users adhere to usage guidelines or restrictions to access the model or implementing safety filters.
- Datasets that have been scraped from the Internet could pose safety risks. The authors should describe how they avoided releasing unsafe images.
- We recognize that providing effective safeguards is challenging, and many papers do not require this, but we encourage authors to take this into account and make a best faith effort.

12. **Licenses for existing assets**

Question: Are the creators or original owners of assets (e.g., code, data, models), used in the paper, properly credited and are the license and terms of use explicitly mentioned and properly respected?

Answer: [Yes]

Justification: All models are cited in Appendix C.

Guidelines:

- The answer NA means that the paper does not use existing assets.
- The authors should cite the original paper that produced the code package or dataset.
- The authors should state which version of the asset is used and, if possible, include a URL.
- The name of the license (e.g., CC-BY 4.0) should be included for each asset.
- For scraped data from a particular source (e.g., website), the copyright and terms of service of that source should be provided.
- If assets are released, the license, copyright information, and terms of use in the package should be provided. For popular datasets, `paperswithcode.com/datasets` has curated licenses for some datasets. Their licensing guide can help determine the license of a dataset.
- For existing datasets that are re-packaged, both the original license and the license of the derived asset (if it has changed) should be provided.
- If this information is not available online, the authors are encouraged to reach out to the asset's creators.

13. **New assets**

    Question: Are new assets introduced in the paper well documented and is the documentation provided alongside the assets?

    Answer: [NA]

    Justification: We do not release new assets.

    Guidelines:

    - The answer NA means that the paper does not release new assets.
    - Researchers should communicate the details of the dataset/code/model as part of their submissions via structured templates. This includes details about training, license, limitations, etc.
    - The paper should discuss whether and how consent was obtained from people whose asset is used.
    - At submission time, remember to anonymize your assets (if applicable). You can either create an anonymized URL or include an anonymized zip file.

14. **Crowdsourcing and research with human subjects**

    Question: For crowdsourcing experiments and research with human subjects, does the paper include the full text of instructions given to participants and screenshots, if applicable, as well as details about compensation (if any)?

    Answer: [NA]

    Justification: We do not use crowdsourcing or human subjects.

    Guidelines:

    - The answer NA means that the paper does not involve crowdsourcing nor research with human subjects.
    - Including this information in the supplemental material is fine, but if the main contribution of the paper involves human subjects, then as much detail as possible should be included in the main paper.
    - According to the NeurIPS Code of Ethics, workers involved in data collection, curation, or other labor should be paid at least the minimum wage in the country of the data collector.

15. **Institutional review board (IRB) approvals or equivalent for research with human subjects**

    Question: Does the paper describe potential risks incurred by study participants, whether such risks were disclosed to the subjects, and whether Institutional Review Board (IRB) approvals (or an equivalent approval/review based on the requirements of your country or institution) were obtained?

    Answer: [NA]

    Justification: We do not use crowdsourcing or human subjects.

Guidelines:

- The answer NA means that the paper does not involve crowdsourcing nor research with human subjects.
- Depending on the country in which research is conducted, IRB approval (or equivalent) may be required for any human subjects research. If you obtained IRB approval, you should clearly state this in the paper.
- We recognize that the procedures for this may vary significantly between institutions and locations, and we expect authors to adhere to the NeurIPS Code of Ethics and the guidelines for their institution.
- For initial submissions, do not include any information that would break anonymity (if applicable), such as the institution conducting the review.

16. **Declaration of LLM usage**

Question: Does the paper describe the usage of LLMs if it is an important, original, or non-standard component of the core methods in this research? Note that if the LLM is used only for writing, editing, or formatting purposes and does not impact the core methodology, scientific rigorousness, or originality of the research, declaration is not required.

Answer: [Yes]

Justification: We study how graph-structured knowledge is represented in various LLMs.

Guidelines:

- The answer NA means that the core method development in this research does not involve LLMs as any important, original, or non-standard components.
- Please refer to our LLM policy (https://neurips.cc/Conferences/2025/LLM) for what should or should not be described.