Graph of Experts: Trainable Graphs for Compositional Reasoning*

Michael Vaden
(mvaden6@gatech.edu)

Georgia Institute of Technology, Atlanta, USA

abstract

0 Introduction

-hierarchical problems require systems that adapt across levels of abstraction

-vesuvius scrolls interesting because of diversity of challenges

-conventional deep networks such as UNets or transformers apply the same operations to every region, lacking mechanisms to dynamically compose specialized behaviors

-reframe modular neural routing as a learnable graph

-each node goes with an expert trained for distinct subtask

-in the vesuvius case, geometry reconstruction, fiber orientation, ink segmentation

-directed nodes encode transition policies governing how information propagates

-GoE learns structured pathways through the expert graph, discovering intermediate representations to solve complex, multi-scale problems

-unifies specialization and coordination

-gating mechanism dynamically selects experts and edge transitions

-allows for self-organization into meaningful workflows

-Add experiment info [**?**].

1 Background and Related Work

-Recent progress in adaptive architectures explores dividing neural computation into specialized components

-MoE paradigm trains multiple experts in a specific data regime, while a gating network selects which to activate per input

-Improves efficiency but remains flat due to one pass

-Google's mixture of recursions introduces recursive expert calls, anbling dynamic reasoning chains, these are typically linear or sequential, optimized for symbolic or temporal reasoning tasks

-They lack explicit modeling of relationships between experts themselves, for instance, how information should transition between specialists handling distinct subproblems

-Each edge encodes transition policies

-Gating mechanism operates not as a simple router but as a graph traversal policy

-Trained to discover efficient and semantically meaningful pathways across experts

-Learns compositional workflows rather than isolated specializations

-Mention vesuvius relation

2 Graph of Experts

In this work we introduce Graph of Experts (GoE) as a novel architecture for compositional and structured reasoning across interconnected neural specialists. Unlike traditional mixture-of-experts models that operate in a single routing step, GoE organizes experts as nodes within a directed graph whose edges encode learnable transition policies. Each expert learns a specialized transformation, while the graph topology and routing dynamics determine how information flows and recombines across experts.

2.1 Problem

We consider GoE as a unified model for the coupled tasks presented by the Vesuvius Challenge of geometry unwrapping and ink detection.

2.2 Input Parameterization

The input stem acts as a modality adapter for transforming input data into standardized latent and tokenized representations. Given an input batch, $x \in \mathbb{R}^{[B,C,\ldots]}$, with

*To whom correspondence should be addressed Tel: +1-240-381-2383; Fax: +1-202-508-3799; e-mail: info@schtm.org

batch size B, channel count C, and remaining modality-dependent dimensions, x should be mapped into the latent representation

$$f_0 = \phi(x; \theta_s) \tag{1}$$

where ϕ is a parameterized feature extractor (e.g. convolutional stack for spatial data, temporal encoder for sequences). f_0 is then tokenized by partitioning or sampling the representation into discrete tokens

$$t_i = W_t \psi(f_0, \Omega_i) + b_t, \qquad T = \{t_1, \ldots, t_N\} \tag{2}$$

where ψ extracts the data slice (e.g. voxels, patches, temporal sequences) corresponding to the i-th token and W_t projects it into a shared embedding space of dimension d. This process yields a modality-independent token set $T \in \mathbb{R}^{B,N,d}$. By decoupling the feature extraction and tokenization mechanisms from any specific input structure, the input stem acts as a general adapter capable of ingesting multimodal data into a coherent token space.

In addition to token generation, the input stem produces a compact set of auxiliary routing features

$$a_i = g_{\text{aux}}(f_0[\Omega_i]) \tag{3}$$

which summarize routing-relevant characteristics within each token's receptive field and provide the graph router with structural cues for expert selection. g_{aux} should emit a standardized descriptor $a_i \in \mathbb{R}^{d_a}$ for every token. For example, in spatial domains such as the Vesuvius scrolls, g_{aux} may be implemented as a lightweight 3D convolutional projection or a pooled neighborhood encoder that captures local curvature, gradient, or anisotropy patterns. In contrast, for sequential or linguistic data, it could take the form of an attention-based or temporal pooling mechanism that encodes syntactic or positional dependencies.

2.3 Encoder

Given input stem output (T, a), the encoder operates on the token set $T = \{t_1, \ldots, t_N\}$ and their associated auxiliary routing features $a = \{a_1, \ldots, a_N\}$. The encoder learns contextual dependencies among tokens while maintaining domain invariance, enabling shared reasoning across heterogeneous modalities. Formally, the encoder applies a sequence of permutation-invariant transformations

$$T' = \Phi(T; \theta_e) \tag{4}$$

where Φ denotes a stack of self-attention and feed-forward blocks of the form LayerNorm $\rightarrow$ MultiHeadAttention(Q,K,V) $\rightarrow$ Residual $\rightarrow$ FeedForward $\rightarrow$ Residual

Each attention head computes

$$\text{Attn}(Q,K,V) = \text{softmax}\left(\frac{QK^\top}{\sqrt{d_k}}\right)V \tag{5}$$

allowing the model to dynamically weight relationships among tokens based on learned similarity.

To ensure routing signals evolve alongside the contextual embeddings, the auxiliary features are also transformed through a lightweight path

$$a' = f_{\text{aux}}(a, T') \tag{6}$$

where f_{aux} may consist of a shallow projection or cross-attention mechanism that refines routing features using the updated token context. This coupling preserves alignment between token semantics and routing behavior, allowing downstream routing modules to operate on context-aware features.

The modality-agnostic encoder thus functions as the system's universal relational core, encoding token and routing dependencies in a manner invariant to the input modality. The resulting (T', a') pair provides a unified, enriched representation for subsequent graph routing and expert specialization.

2.4 Graph Router

The graph router defines a directed graph

$$\mathcal{G} = (\mathcal{V}, \mathcal{E}), \qquad |\mathcal{V}| = M \tag{7}$$

where each node $v_m \in \mathcal{V}$ corresponds to an expert module, and each directed edge represents learned dependencies between experts. For each input token t_i' and its auxiliary routing features a_i' is transformed into combined descriptor $r_i = [t_i'; a_i]$, used to compute an entry distribution over experts

$$p_i = \text{softmax}(W_r r_i + b_r), p_i \in \mathbb{R}^M$$

where $p_{i,m}$ indicates how strongly token i is routed to expert m. Once tokens enter the graph, they propagate through connected experts according to the learned edge weights A_{mn}

$$h_m^{l+1} = \rho\Big(\sum_{n \in \mathcal{N}(m)} A_{mn} W_n h_n^l\Big)$$

where A may be static (learned during training) or dynamically updated via attention over expert embeddings. This enables experts to share context, refine intermediate representations, or coordinate when multiple experts specialize in related subspaces. Routing can thus be interpreted as probabilistic traversal through the expert graph based on three principles:

1. *Auxiliary features* bias the entry points.
2. *Graph topology* defines how expertise flows among related experts.
3. *Router distribution* p_i determines which paths are activated for each token.

This formulation allows computation to be selective, structured, and interpretable: experts form the nodes of a learned relational system, and data tokens dynamically navigate it based on both semantic content and contextual metadata.

2.5 Experts

Each node $v_m \in \mathcal{V}$ of the expert graph represents a specialized processing unit trained to perform a distinct trans-

formation on routed token representations. Collectively, these experts form a distributed reasoning system in which specialization emerges from data-driven routing rather than manual assignment. Formally, every expert implements a parameterized function

$$h_m^{l+1} = f_m(h_m^l; \theta_m)$$

where h_m^l is the set (or mean) of token embeddings arriving at node m during layer l. The internal structure of f_m is modality-independent—it can be a small transformer block, an MLP, a convolutional unit, or any lightweight operator suited to the representation space.

Routing probabilities from Section 3.3 determine which tokens reach each expert:

$$\widetilde{h}_m = \sum_i p_{i,m} W_p r_i$$

aggregating routed inputs according to their gating weights $p_{i,m}$. Experts update their local states through a combination of internal processing and graph message passing:

$$h_m^{l+1} = f_m(\widetilde{h}_m; \theta_m) + \sum_{n \in \mathcal{N}(m)} A_{mn} W_c h_n^l$$

where A_{mn} encodes inter-expert communication strength and W_c projects incoming context from neighboring experts.

During training, load-balancing and entropy regularizers encourage even utilization and discourage collapse to a single dominant expert:

$$\mathscr{L}_{\text{balance}} = \lambda_b \sum_m (\frac{1}{B} \sum_i p_{i,m} - \frac{1}{M})^2$$

$$\mathscr{L}_{\text{entropy}} = -\lambda_e \sum_i \sum_m p_{i,m} \log p_{i,m}$$

These terms ensure diverse specialization while maintaining differentiability for end-to-end learning.

The expert layer therefore acts as a distributed knowledge substrate: each expert learns to focus on a coherent subset of representations, yet coordination through the graph preserves global consistency. This structure supports both horizontal specialization (different skills in parallel) and vertical specialization (multi-stage refinement across connected experts).

TODO Token recombination

To enable progressive reasoning and self-correction, the architecture employs a recursive refinement mechanism in which expert activations are iteratively updated based on their own outputs and feedback from neighboring experts. Rather than processing each token once, the system performs multiple refinement cycles

$$H^{k+1} = \mathscr{F}(H^k; \theta_{\text{exp}}; \theta_{\text{router}}), k = 0, \ldots K-1$$

where $H^k = \{h_1^k, \ldots h_M^k\}$ represent all expert states after iteration k. During each cycle three actions are performed:

1. *Re-routing*: Updated token embeddings are re-evaluated by the graph router using current auxiliary cues to adjust their expert assignments

   $$p_i^{k+1} = \text{softmax}(W_r[r_i^k] + b_r)$$

   allowing the computation graph to evolve dynamically as uncertainty decreases.
2. *Expert update*: Each expert processes its new incoming messages

   $$h_i^{k+1} = f_m(\sum_i p_{i,m}^{k+1} W_p r_i^k, h_{\mathcal{N}(m)}^k; \theta_m)$$

   combining local evidence with context from adjacent experts.
3. *Convergence check*: A lightweight confidence head evaluates per-token uncertainty

   $$u_i^k = \rho(W_u r_i^k)$$

   and recursion halts when $||u^k - u^{k-1}||_2 < \varepsilon$ or after a fixed depth K.

This iterative process lets the system focus compute where ambiguity remains—tokens with stable expert assignments converge quickly, while uncertain regions receive additional refinement passes.

The recursive design unifies iterative inference and conditional computation: information flows repeatedly through the expert graph, allowing specialists to exchange context and correct earlier approximations. In practice, this produces smoother, more consistent outputs while maintaining computational efficiency through early stopping.

2.6 Decoders

3 SUMMARY

Filtering an audio signal with an allpass filter does not usually have a major effect on the signal's timbre. The allpass filter does not change the frequency content of the signal, but only introduces a phase shift or delay. Audibility of the phase distortion caused by an allpass filter in a sound reproduction system has been a topic of many studies, see, e.g., [**?**], [**?**]. In this paper, we investigate audio effects processing using high-order allpass filters that consist of many cascaded low-order allpass filters. These filters have long chirp-like impulse responses. When audio and music signals are processed with such a filter, remarkable changes are obtained that are similar to the spectral delay effect [**?**], [**?**].

4 CONCLUSION

Filtering an audio signal with an allpass filter does not usually have a major effect on the signal's timbre. The allpass filter does not change the frequency content of the signal, but only introduces a phase shift or delay. Audibility of the phase distortion caused by an allpass filter in a sound reproduction system has been a topic

of many studies, see, e.g., [**?**], [**?**]. In this paper, we investigate audio effects processing using high-order allpass filters that consist of many cascaded low-order allpass filters. These filters have long chirp-like impulse responses. When audio and music signals are processed with such a filter, remarkable changes are obtained that are similar to the spectral delay effect [**?**], [**?**]. Note that articles might have a digital object identifier [**?**].

5 ACKNOWLEDGMENT

This research was conducted in fall 2008 when Vesa Välimäki was a visiting scholar at CCRMA, Stanford University. His visit was financed by the Academy of Finland (project no. 126310). The authors would like to Dr. Henri Penttinen for his comments and for the snare drum sample used in this work.

APPENDIX

Filtering an audio signal with an allpass filter does not usually have a major effect on the signal's timbre. The allpass filter does not change the frequency content of the signal, but only introduces a phase shift or delay. Audibility of the phase distortion caused by an allpass filter in a sound reproduction system has been a topic of many studies, see, e.g., [**?**], [**?**].

$$\phi(\omega) = -\omega + 2\arctan\left(\frac{a_1 \sin\omega}{1 + a_1 \cos\omega}\right) \qquad (1)$$

In this paper, we investigate audio effects processing using high-order allpass filters that consist of many cascaded low-order allpass filters. These filters have long chirp-like impulse responses. When audio and music signals are processed with such a filter, remarkable changes are obtained that are similar to the spectral delay effect [**?**], [**?**].

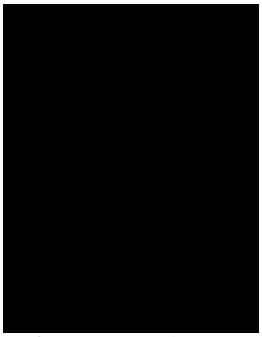
NOMENCLATURE

a_c = condensation coefficient condensation coefficient condensation coefficient condensation coefficient

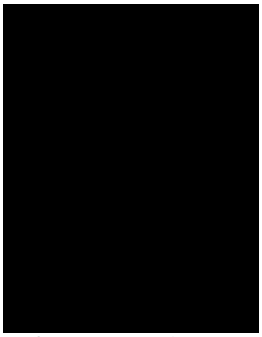
TLR = Toll-like receptor

PAMPs = pathogen-associated molecular patterns condensation coefficient condensation

THE AUTHORS



A1firstname A1lastname



A2firstname A2lastname

A1firstname A1lastname is professor of audio signal processing at Helsinki University of Technology (TKK), Espoo, Finland. He received his Master of Science in Technology, Licentiate of Science in Technology, and Doctor of Science in Technology degrees in electrical engineering from TKK in 1992, 1994, and 1995, respectively. His doctoral dissertation dealt with fractional delay filters and physical modeling of musical wind instruments. Since 1990, he has worked mostly at TKK with the exception of a few periods. In 1996 he spent six months as a postdoctoral research fellow at the University of Westminster, London, UK. In 2001-2002 he was professor of signal processing at the Pori School of Technology and Economics, Tampere University of Technology, Pori, Finland. During the academic year 2008-2009 he has been on sabbatical and has spent several months as a visiting scholar at the Center for Computer Research in Music and Acoustics (CCRMA), Stanford University, Stanford, CA. His research interests include musical signal processing, digital filter design, and acoustics of musical instruments. Prof. Välimäki is a senior member of the IEEE Signal Processing Society and is a member of the AES, the Acoustical Society of Finland, and the Finnish Musicological Society. He was the chairman of the 11th International Conference on Digital Audio Effects (DAFx-08), which was held in Espoo, Finland, in 2008.

●

A2firstname A2lastname is a consulting professor at the Center for Computer Research in Music and Acoustics (CCRMA) in the Music Department at Stanford University where his research interests include audio and music applications of signal and array processing, parameter estimation, and acoustics. From 1999 to 2007, Abel was a co-founder and chief technology officer of

the Grammy Award-winning Universal Audio, Inc. He was a researcher at NASA/Ames Research Center, exploring topics in room acoustics and spatial hearing on a grant through the San Jose State University Foundation. Abel was also chief scientist of Crystal River Engineering, Inc., where he developed their positional audio technology, and a lecturer in the Department of Electrical Engineering at Yale University. As an industry consultant, Abel has worked with Apple, FDNY, LSI Logic, NRL, SAIC and Sennheiser, on projects in professional audio, GPS, medical imaging, passive sonar and fire department resource allocation. He holds Ph.D. and M.S. degrees from Stanford University, and an S.B. from MIT, all in electrical engineering. Abel is a Fellow of the Audio Engineering Society.