GoE (Graph-of-Experts): Trainable Graphs for Hierarchical Multi-Scale Reasoning

Michael Vaden[1,2]

[1] Independent Researcher, Atlanta, USA
[2] College of Computing, Georgia Institute of Technology, Atlanta, USA
*corresponding author: mvaden6@gatech.edu, michaelvaden.mjv@gmail.com

Keywords: Mixture of Experts, Graph Networks, Transformers

Abstract.
-propose a Graph of Experts (GoE) architecture
-general framework for adaptive, compositional problem-solving
-each expert represents a specialized subnetwork
-edges between experts are learnable pathways encoding how info should flow
-gating policy learns which experts to invoke and how to traverse the expert graph
-gating enables dynamic routing, feedback, and recombination of partial results
-single pass -¿ structured reasoning
-blueprint for modern hierarchical intelligence
-efficient computation along graphs
-could coordinate reasoning across multi-modal domains
-need experiment results

1 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

2 Background

-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

3 Model Architecture

Here we formalize GoE as a modular pipeline. We cleanly separate input stems, modality-sepecific adapters that transform raw data into tokens with positions/scale, from a task-agnostic encoder that adds local/global context and emits per-token content embeddings h (for experts) and routing features e (for the graph router). The GoE core then performs sparse traversal over a library of lightweight experts under a learned graph router, allocating computation by difficulty and recording path provenance. Task heads are pluggable and minimal (used only for supervision/inference), so stems and heads change with the domain while the GoE core is identical.

3.1 Input Stem

Given a raw input x, the input stem should produce some sequence of tokens, $T \in \mathbb{R}^{B \times N \times d}$, as well as the auxiliary routing features for the graph router, $A \in \mathbb{R}^{B \times N \times d_a}$. The input stem is the only component of the system that is modality-specific, as it is the modality adapter for the problem.

3.2 Encoder

The encoder provides global context for the auxiliary routing data with modality-specific methods. Given input from the input stem, the encoder produces multi-level feature maps, $F^0, \dots F^L$, with F^L being the least dimensional feature map. This is executed by using repeatable G-Blocks with residuals. The encoder then installs a slight attention mechanism to ensure multi-modality multi-reasoning. It should be fused minimally via a content embedding. Routing features should also be extracted from the auxiliary routing features.

3.3 Graph Router

Experts $\{f_k\}_{k=1..M}$ are generally trainable neural circuits with parameter efficient adapters for modality hints.

3.4 Experts -include training section

3.5 Traversal

3.6 Task Heads and Losses

Conflict of interests

The authors should declare here any potential conflicts of interests.

Acknowledgments (optional)
The authors would like to thank XXX and YYY for their helpful feedback.

Funding (optional)
This work was supported by the XXX agency (grant number: XXX).

Availability of data and software code (optional and strongly suggested)
Our software code is available at the following URL: XXX.
Our dataset is available at the following URL: XXX.

References