Conditioning the Forward Logit Operator Through Task Evaluation

Bill Cochran

July 30, 2025

Abstract

Language generation can be decomposed into six stages: (1) abstraction of meaning into tokens, (2) finite sampling of these tokens, (3) application of this sampling to a neural network to compute internal activations, (4) projection back into the finite token space, (5) expansion into the infinite token space, allowing for (6) semantic unit selection. We define step (5), the token expansion step, as the *forward logit operator*. In this work, we propose a mechanism for conditioning this operator with basic task evaluation.

1 Introduction

In classical neural networks, so the intuition goes, depth corresponds to abstraction. Each successive layer composes lower-order features into increasingly complex representations, approximating higher-order logical predicates over the input domain. While this phenomenon is widely observed empirically, it remains under-characterized in terms of the projection geometry between internal representations and output behavior.

In this work, we examine how these high-level internal representations—what we term *concept nodes*—can be harnessed to modulate the output distribution in a structured and theoretically principled manner. Rather than treating the final-layer logits as a fixed projection from hidden state to token space, we view them as an intermediate surface: a mutable space in which meaning, preference, and structure may be reintroduced through targeted transformations.

We propose a framework in which a secondary learner—such as a convolutional network, recurrent architecture, or multilayer perceptron—operates on the concept node activations to produce a transformation in logit space. This transformation serves as a dynamic filter, applied to the base logits either multiplicatively or additively, altering the output distribution in a way conditioned on higher-order internal structure.

This conditioning mechanism may be interpreted as a second-order logic step, applied not over input tokens but over internal concepts. It augments the forward pass with an auxiliary computation whose role is to shape the model's output trajectory according to semantic content encoded at depth. Because this transformation is expressed in logit space, it interfaces naturally with token selection and decoding, while remaining disentangled from the core model weights.

We formalize this view by treating the final logit vector as an object subject to transformation: not merely the result of a projection, but the input to a secondary operator that encodes task-specific logic, temporal preference, or structural bias. This layered approach provides a means of modulating language generation without altering the underlying network, enabling a clean separation between representation and control.

Finally, this perspective suggests a natural interface for injecting structured feedback into a language model: the logit space itself. In conventional usage, a language model must infer, *a priori*, whether it is performing well—relying solely on implicit cues from the prompt and its own training. There is no inherent mechanism for runtime evaluation or correction of its current behavior in light of task success.

By contrast, conditioning the output logits via a transformation grounded in concept space provides a direct channel through which external or internal feedback *a posteriori*. Rather than backpropagating through the entire model, we can apply lightweight corrections that indicate, for instance, that certain tokens are ineffective or undesirable in the current context. This suggests that task specialization in language models may be achieved through structured conditioning of output logits—mirroring the way human learning refines behavior through iterative feedback and instruction. Practice makes perfect.

2 Related Work

In [1], we develop a set of linear tensor operators that describe the language token generation problem.

References

[1] Bill Cochran. On the limits of hierarchically embedded logic in classical neural networks. *arXiv preprint arXiv:2507.20960*, 2025.