When '+' Means '-'
Probing Arithmetic Circuits Under Symbol Redefinition

Aditeya Baral   Allen George Ajith

Courant Institute of Mathematical Sciences, New York University

Motivation

Prior work shows that when LLMs perform arithmetic, they reliably use a consistent **internal computation pathway**, also known as a **neural circuit**. Arithmetic symbols such as **+**, **−**, and × behave as strongly typed operators that activate these learned circuits.

Question: What happens when we *redefine operators in-context*? When we tell the model "**+** means ×" through few-shot examples, does it:

1. Reuse the addition circuit with modified inputs (*semantic* understanding), or
2. Activate a completely different circuit (*syntactic* symbol-binding)?

This reveals whether LLMs treat arithmetic operators as meaningful *semantic primitives* or merely *surface-level tokens* to be overridden by context.

Experimental Setup

Each run consists of a prompt with 8 few-shot expressions that demonstrate the operator's intended behavior. The model predicts the result of a held-out expression using the same operator.
There are two types of runs:

- *Original:* Standard operator semantics
- *Overloaded:* Redefined semantics (e.g., `3 + 4 = 12` implying **+** $\to$ ×)

Operator mappings: Six non-identity mappings over {**+**, **−**, ×}:

$$+\to-,\quad +\to\times,\quad -\to+,\quad -\to\times,\quad \times\to+,\quad \times\to-$$

Model: `meta-llama/Llama-3.3-70B-Instruct`
Dataset: 2,000 prompts per mapping; each operand is sampled $\in\{0,\ldots,9\}$

Original vs Overloaded Accuracy

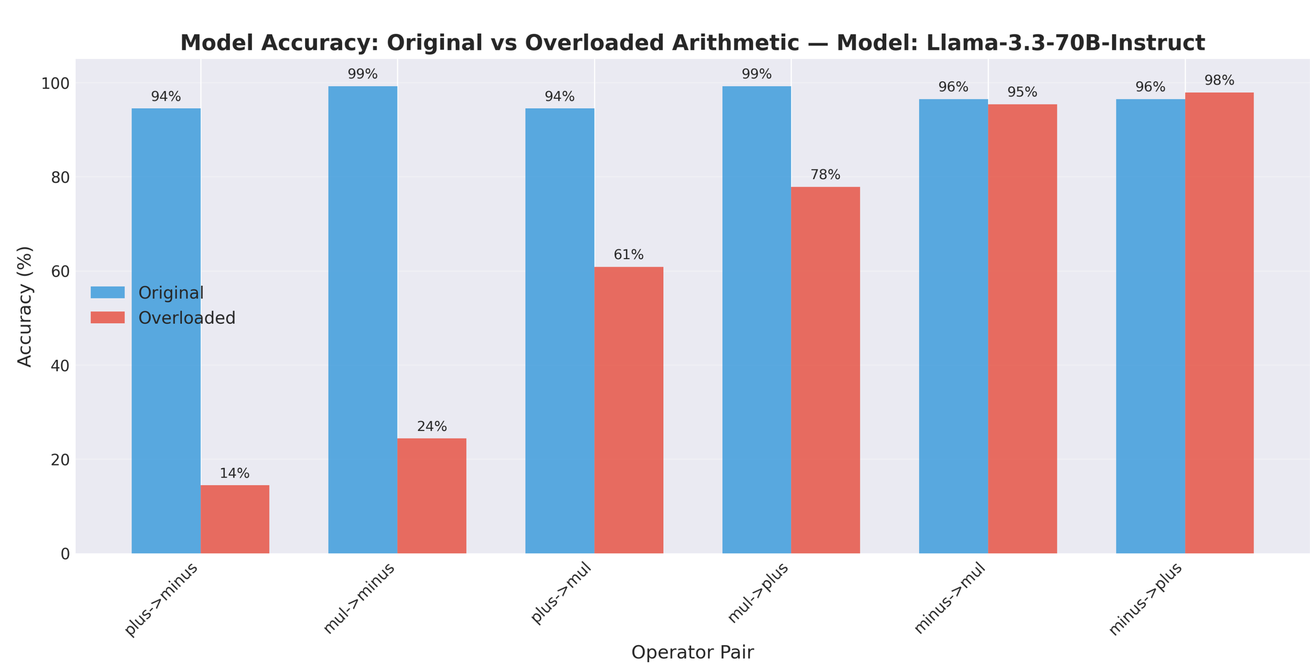

Figure 1. Performance under overloading varies, with $X \to -$ being particularly challenging.

Activation Geometry

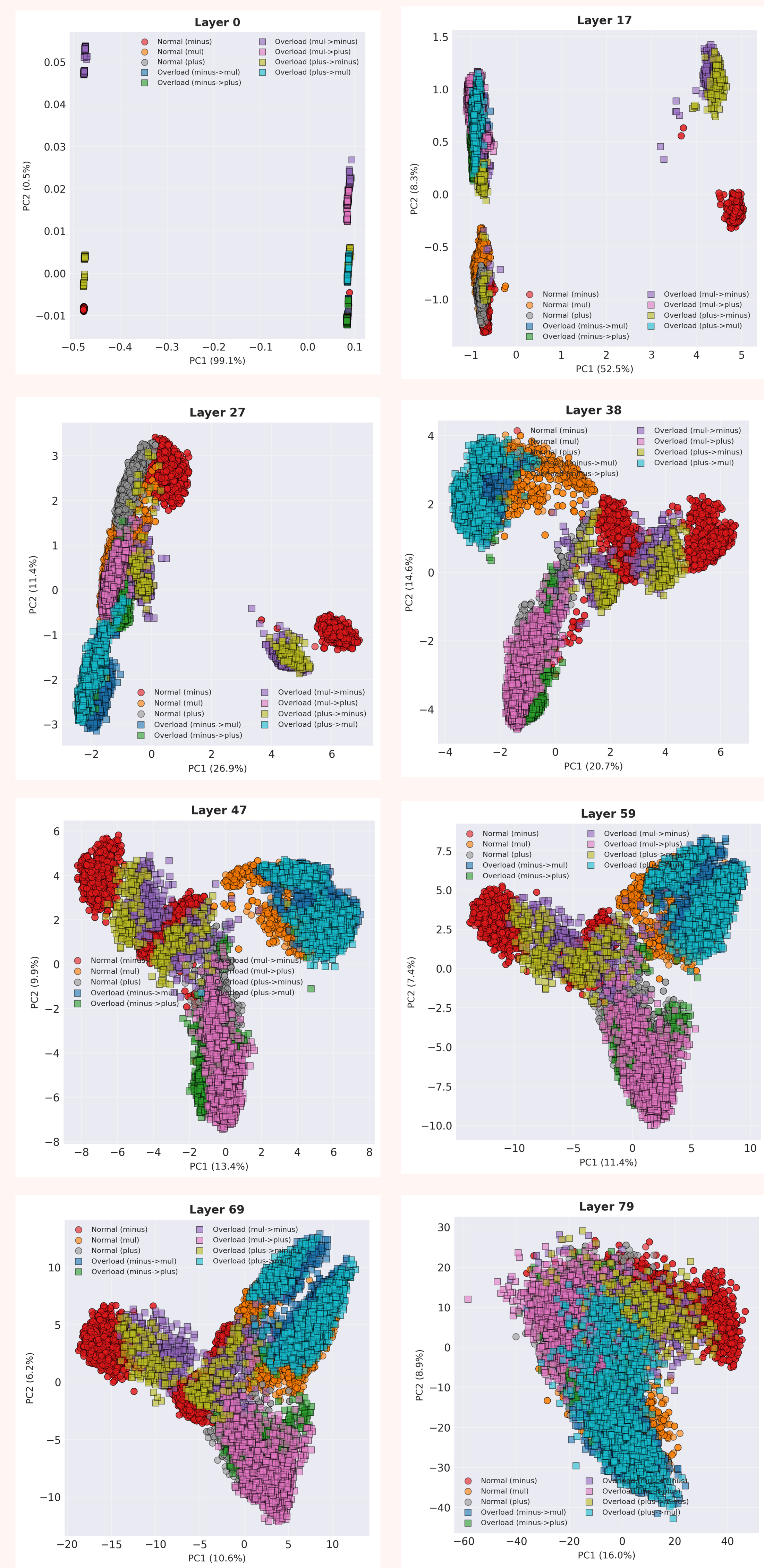

Figure 2. **Evolution of representations induced by overloaded operators ($X \to T$).** Early layers exhibit *representational divergence* with tightly clustered activations grouped by the surface operation X. Across intermediate layers, representations are progressively restructured toward the *target operation* T, forming emerging lobes sharing the same T. These become more compact in deeper layers, indicating *maximal semantic grouping*, before collapsing into a single mass that reflects **late-stage semantic convergence**.

Representational Dynamics

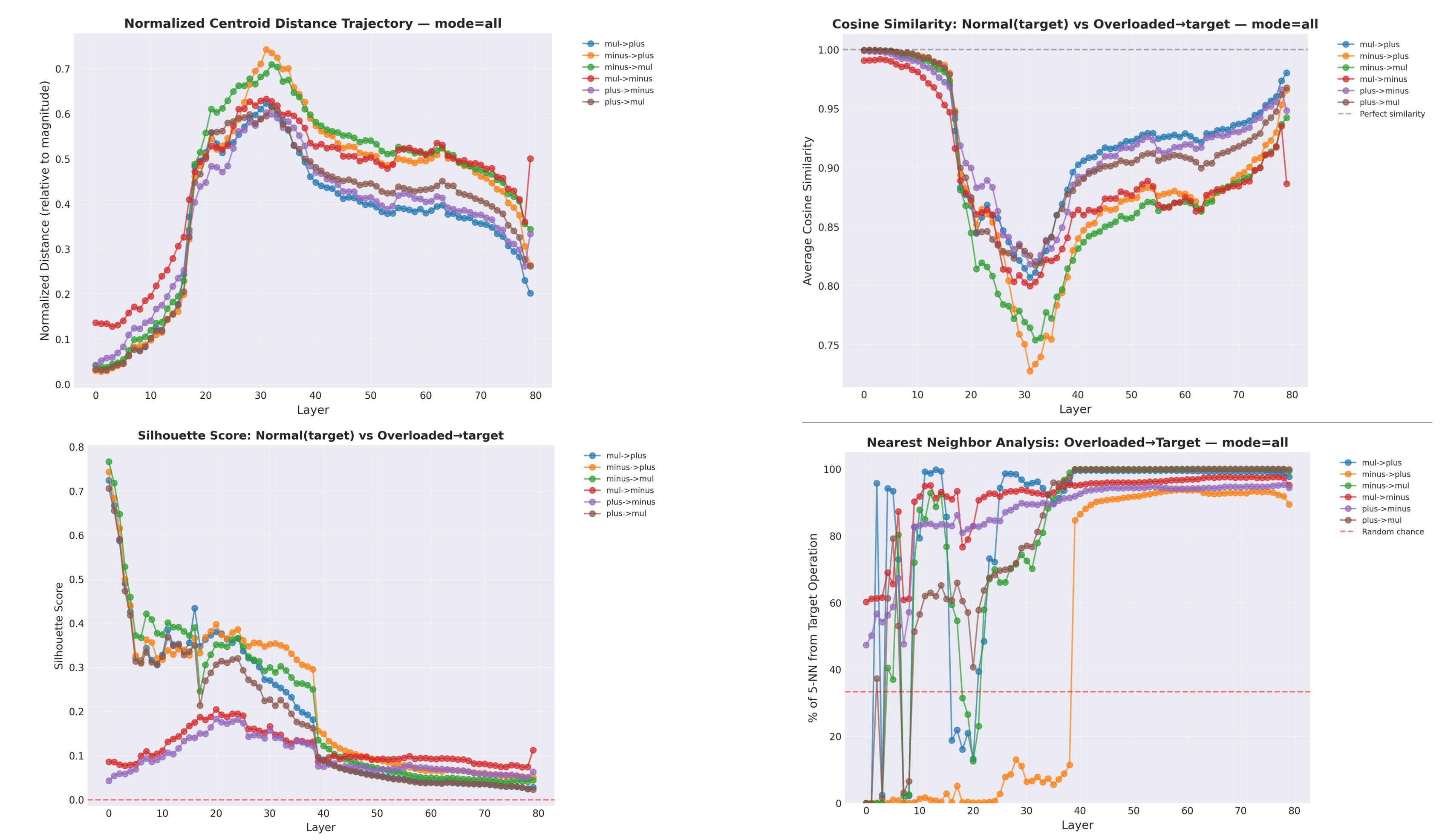

Figure 3. **Layer-wise semantic convergence under operator overloading ($X \to T$).** Representations initially temporarily diverge under operator overloading according to the surface operator X, but progressively reorganize across transformer layers to align with those of the corresponding normal target operation T. This late-layer *alignment* indicates convergence onto a **shared semantic circuit**, supporting semantic reuse through **internal representational remapping** rather than purely syntactic instruction following.

Attention Circuit Dynamics

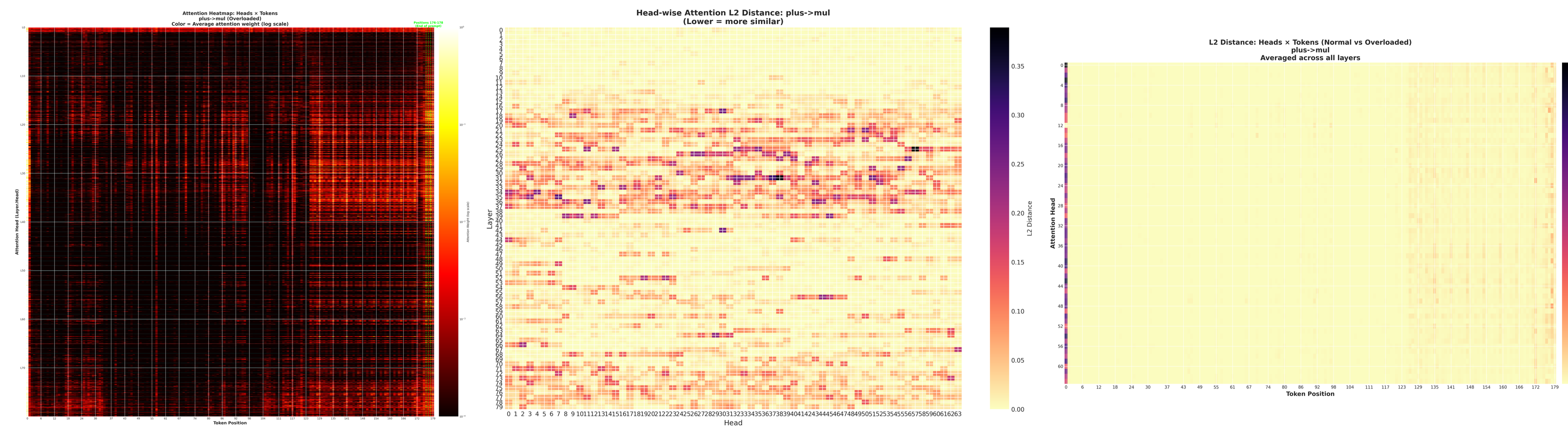

Figure 4. **Attention reconfiguration supporting semantic convergence under operator overloading ($X \to T$).** Consistent with the representational dynamics observed in activations, attention patterns remain largely unchanged in early layers, then undergo structured, layer-localized divergence in mid (15–40) and late (70–80) layers. This reconfiguration is highly selective: differences are concentrated on *few-shot examples* and the *final query*, while earlier tokens remain stable. These token- and layer-specific shifts indicate that attention *reorganizes information flow* to align mid-layer divergence with the target operation, supporting **late-stage semantic convergence**.

ab12057, aa12938 @nyu.edu