LLM Internals: No Syntax Pass – Implications for CPUX Design

Large Language Models (LLMs) do not generate language in two distinct phases — first for meaning, then for grammar. There is no post-processing or correction phase for syntax. Instead, LLMs use a single-phase transformer process: They observe the sequence of tokens in the input context They compute attention across prior tokens to build a probability distribution They predict the *next best token* — one token at a time The appearance of syntactically valid output is purely a side effect of training on vast amounts of well-formed text. There is no internal grammar-checking step or abstract syntax tree verification. **Implication:** Syntax emerges statistically, not logically. For CPUX and Intention Space, this insight is critical. LLMs should not be expected to: Validate logical branching Enforce field completeness Guarantee structural coherence Instead, they should be used for: Generating plausible Intention Phrases and Pulse transitions Extending CPUX flows in a way that "feels" coherent Exploring how prior pulses could influence next-state generation **CPUX Enforces What LLMs Can't:** The Intention Space runtime verifies all logical coherence, trivalent Pulse presence, and PnR gating. LLMs provide language momentum; Intention Space provides structural truth.