LIP: A Universal Latent Injection Protocol for Heterogeneous Model Communication

Cristiano Silva[1]

[1]Independent Research Lab, São Paulo, Brazil
ziwehdafe@gmail.com

November 22, 2025

Abstract

Since 1943, the field of Artificial Intelligence has been defined by the pursuit of teaching machines to think, and subsequently, to speak like humans. Despite the development of sophisticated optimization techniques—Chain-of-Thought, RAG, Vector Databases—we have overlooked a fundamental truth: machines represent intelligence differently than humans. This paper argues that the current standard of inter-model communication (natural language text) is an inefficient, ambiguous, and insecure bottleneck. We illustrate this with the "Dinner Table Paradox": two agents sitting together but forced to exchange handwritten letters to communicate. To resolve this, we introduce the **Latent Injection Protocol (LIP)**, a framework that enables direct, vector-based communication between heterogeneous models. By bypassing the tokenization layer, LIP offers three key advantages: (1) transmission of raw semantic intent rather than ambiguous text; (2) reduced latency; and (3) native data obfuscation, preventing human-readable interception of sensitive reasoning. We validate this protocol by demonstrating the first functional "telepathic" control of a Llama-3-8B model by a TinyLlama-1.1B agent.

1 Introduction

Since 1943, we have tried to teach machines to think like humans. Sometime later, we tried to teach them to speak—like humans. In the decades that followed, we researched and created a myriad of tools and techniques to make this viable, constantly striving to lower costs and improve efficiency.

We created Chain-of-Thought, Retrieval-Augmented Generation (RAG), Vector Databases, Knowledge Graphs... an endless list of techniques to improve accuracy, latency, and the cost of communication in natural language models. But in the midst of this optimization race, we are forgetting something very basic: **machines are still machines** (even if they know how to read and write in English).

1.1 The Dinner Table Paradox

Imagine two human beings, sitting at a dinner table, right next to each other. Now imagine that, to converse during their meal, they are forced to write handwritten letters to one another. This is exactly what we do when we clog the internet by forcing our Large Language Models (LLMs) to send JSON packets for every task that needs to be performed.

Text does not carry *intent*. It carries semantics, and perhaps a brief context. But text is easily *under-* and *over-interpreted*. How many times have you had to repeat a question because a professor didn't understand? Or how many times have you redone a presentation because a Manager or C-Level executive interpreted your statements through a lens you didn't expect?

1.2 The Security Imperative

Furthermore, text is understandable—to humans. This is good for Human-Computer Interaction (HCI), but it is a critical vulnerability for Machine-to-Machine (M2M) interaction. If a human intercepts the network traffic of your enterprise LLM, they gain immediate access to every piece of data, information, and question ever asked to the machine.

As if that weren't enough, even under contracts, the large corporations that own the models effectively own the data. With every prompt, a model is refined—and people (and companies) no longer truly know what "confidential" or "secret" information means. All this because text is easily kidnapped, intercepted, and interpreted.

1.3 Our Contribution: The LIP Protocol

With this in mind, this work presents our contribution to the field: a universal communication protocol between natural language models. It performs three tasks with mastery:

1. **LIP Protocol Implementation:** Packets follow a fixed, vector-based standard, independent of the models used.
2. **Energy Matching Mechanism:** A calibration system to align signal magnitudes.
3. **Heterogeneous Transfer Validation:** Successfully bridging the gap between distinct architectures ($2048d \rightarrow 4096d$).

What does this imply for the future of AI?

- **Intent Preservation:** Vectors carry far more than text; they carry the intent, the semantics, and the context directly from the hidden states of the reasoning process.
- **Latency Reduction:** Vector-to-Vector (V2V) communication is inherently faster than Text-to-Text (T2T) generation and parsing.
- **Native Obfuscation:** Sending vectors adds a layer of obfuscation directly into the model's operation. This decreases external dependency and increases total application security, making data leaks significantly harder.
- **Encryption Potential:** It is possible to apply mathematical cryptography on top of these packets, making it impossible to discover the exact content of a '.lip' packet. Sensitive information becomes mathematically secure.
- **Universal Scalability:** The protocol allows connecting tiny models—as we did with Llama 1.1B—to massive models. This validates the potential of using a 1.1B model to trigger text generation in a super-model (e.g., 80B parameters), preserving semantic intent while expanding logical reasoning capabilities.

2 Methodology

We propose a framework consisting of an asymmetric Dual-Encoder Adapter and the Latent Injection Protocol (LIP).

2.1 Asymmetric Dual-Encoder Architecture

To bridge the representational gap between a source model M_S (TinyLlama-1.1B, $d_S = 2048$) and a target model M_T (Llama-3-8B, $d_T = 4096$), we introduce a bottleneck adapter.

$$\mathbf{z} = \sigma(\text{LN}(\mathbf{W}_E \cdot \mathbf{h}_S + \mathbf{b}_E)) \tag{1}$$

This effectively compresses the "intent" into a universal format.

2.2 Energy Matching Calibration

To address the "handwritten letter" inefficiency, we ensure the signal is mathematically calibrated:

$$\mathbf{v}_{calibrated} = \mathbf{v}_{inj} \cdot \frac{E_{ref}}{||\mathbf{v}_{inj}||_2 + \varepsilon} \tag{2}$$

3 Experiments

We validated the "Universal Scalability" claim by pairing TinyLlama (Consumer/Edge) with Llama-3 (Server/Cloud). The adapter successfully translated abstract intents (e.g., "define a python sum function") into complex code generation without a single token of text being exchanged between the agents.

4 Conclusion

We have demonstrated that we do not need to force machines to speak human language to coordinate effectively. By adopting the Latent Injection Protocol, we move from an era of ambiguous, insecure text exchange to an era of precise, obfuscated, and efficient vector communication.

References